GE Fanuc Automation

Programmable Control Products

Motion Mate™ DSM302 for Series 90™-30 PLCs

User's Manual

GFK-1464C December 2002

GFL-002

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

Content of This Manual

This manual describes the Motion Mate DSM302 (Release 1.40), which is a complete integrated motion control system. The Motion Mate DSM302 is an intelligent, fully programmable, motion control option module for the Series 90-30 Programmable Logic Controller (PLC). The Motion Mate DSM302 allows a PLC user to combine high performance motion control with PLC logic solving functions in one integrated system. This manual consists of nine chapters and seven appendices.

Chapter 1. Product Overview: This chapter provides an overview of the hardware and software used to set up and operate a Motion Mate DSM302 motion control system.

Chapter 2. Getting Started: This chapter provides an introduction to the Motion Mate DSM302 motion control system through a start-up guide that outlines the steps required to operate or jog your motion controller and servo. The startup guide is intended for the new user, but will help all users to get their system *up and running* in a short time.

Chapter 3. Installing and Wiring the DSM302: This chapter provides the installation and wiring information required for your Motion Mate DSM302 motion control system.

Chapter 4. Configuring the DSM302: This chapter explains how to configure the Motion Mate DSM302 using the Logicmaster 90-30 programming software configuration function. Note that the later releases of the graphical Control configuration software also support the DSM302.

Chapter 5. Motion Mate DSM302 to PLC Interface: This chapter describes the %I, %AI, %Q, and %AQ data that is transferred between the Motion Mate DSM302 and the Series 90-30 PLC CPU.

Chapter 6. Non-Programmed Motion: This chapter describes the six different ways that non-programmed motion is generated.

Chapter 7. Programmed Motion: This chapter describes how the DSM302 executes program motion commands sequentially in a block-by-block fashion in a selected program.

Chapter 8. Follower Motion: This chapter describes how follower motion is executed by the DSM302.

Chapter 9. Combined Follower and Commanded Motion: This chapter describes combined motion, which consists of follower motion commanded from a master axis combined with specific internally commanded motions.

Appendix A. Error Reporting: This appendix describes the errors reported by the %AI module status code and axis error code words.

Appendix B. Parameter Download Using a COMM_REQ: This appendix describes how to use the COMM_REQ function block to load Data Parameter Memory from the PLC to the DSM302.

Appendix C. Position Feedback Devices: This appendix provides information needed to use Fanuc serial encoders and incremental quadrature encoders with the DSM302.

Appendix D. Startup and Tuning a GE Fanuc Digital Servo System: This appendix provides a procedure for starting up and tuning a GE Fanuc digital servo system.

Appendix E. Using the APM Motion Programmer Status Screen with the DSM302: This appendix describes how the APM Motion programmer may be used to monitor the operation of a DSM302 module.

Appendix F. Updating Firmware in the DSM302: This appendix describes the procedure for updating firmware in Flash memory in the DSM302**.**

Appendix G. Answers to Frequently Asked Questions: This appendix contains answers to the most frequently asked motion questions.

Related Publications

For more information, refer to the following publications:

Series 90™-30 Programmable Controller Installation Manual: GFK-0356
Logicmaster™ 90 Series 90™-30/20/Micro Programming Software User's Manual: GFK-0466
Series 90™-30/20/Micro Programmable Controller's Reference Manual: GFK-0467
Installation Requirements for Conformance to Standards: GFK-1179
Series 90™ PLC APM Programmer's Manual: GFK-0664
α Series Servo Amplifier (SVU) Descriptions Manual: GFZ-65192EN/01
Control Motor Amplifier α Series (SVM): GFZ-65162E/02
α Series Servo Motor Manuals: GFZ-65165E/01, GFZ-65150E/02, GFZ-65142E/02
β Series Servo Product Specification Guide: GFH-001
SL Series Servo User's Manual; GFK-1581

Contents

Contents

Contents

Chapter 1 Product Overview

The Motion Mate DSM302 is a high performance, easy-to-use, 2-axis motion control module which is highly integrated with the Series 90-30 PLC logic solving and communications functions.

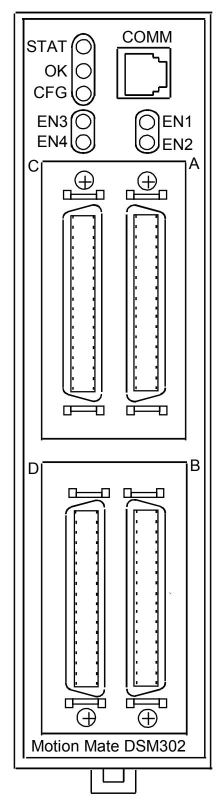
The DSM302 operates in two primary control loop configurations:

- Standard Mode
- Follower Mode

Servo Types Supported

- Digital – All firmware releases of the DSM302 support the GE Fanuc digital servo amplifiers and motors.
- Analog – Starting with firmware release 1.40, support for analog servos was added to the DSM302. The GE Fanuc SL Series analog servos are documented in the *SL Series Servo User's Manual*, GFK-1581.

Features of the Motion Mate DSM302



High Performance

- Digital Signal Processor (DSP) provides Vector Control of GE Fanuc AC Servos
- Block Processing time under 5 milliseconds
- Velocity Feed forward and Position Error Integrator to enhance tracking accuracy
- High resolution of programming units
  - Position: -8,388,608...+8,388,607 User Units
  - Velocity: 1 ... 8,388,607 User Units/sec
  - Acceleration: 1 … 134,217,727 User Units/sec/sec

Easy to Use

- Simple and powerful Motion Program instruction set
- Simple 1- or 2-axis motion programs with synchronized block start
- Program support for a short motion program which can be created in Logicmaster 90-30/20/Micro configuration software
- Non-volatile storage for 10 programs and 40 subroutines
- User scaling of programming units (User Units)
- Flash update of system firmware
- Generic programming using command parameters as operands for Acceleration, Velocity, Move, and Dwell Commands
- Configured with Logicmaster 90-30/20/Micro software (version 8.02 or later)
- Automatic Data Transfer between PLC tables and DSM302 without user programming
- Ease of I/O connection with factory cables and terminal blocks as well as a serial port for connecting programming devices

Versatile I/O

- Control of GE Fanuc Digital AC servos
- +/-10volt Velocity Command analog output
- Home and overtravel switch inputs for each Servo Axis
- Two Position Capture Strobe Inputs for each Position Feedback Input
- 5v , 24v and analog I/O for use by PLC
- Incremental Quadrature Encoder input on each axis for Encoder/Analog mode
- Incremental Quadrature Encoder input for Follower Master axis
- 13 bit Analog Output can be controlled by PLC or used as Digital Servo Tuning monitor

Section 1: Motion System Overview

The DSM302 is an intelligent, fully programmable, motion control option module for the Series 90-30 Programmable Logic Controller (PLC). The DSM302 allows a PLC user to combine high performance motion control with PLC logic solving functions in one integrated system. The figure below illustrates the hardware and software used to set up and operate a servo system. This section will discuss briefly each system element to provide an overall understanding of system operation.

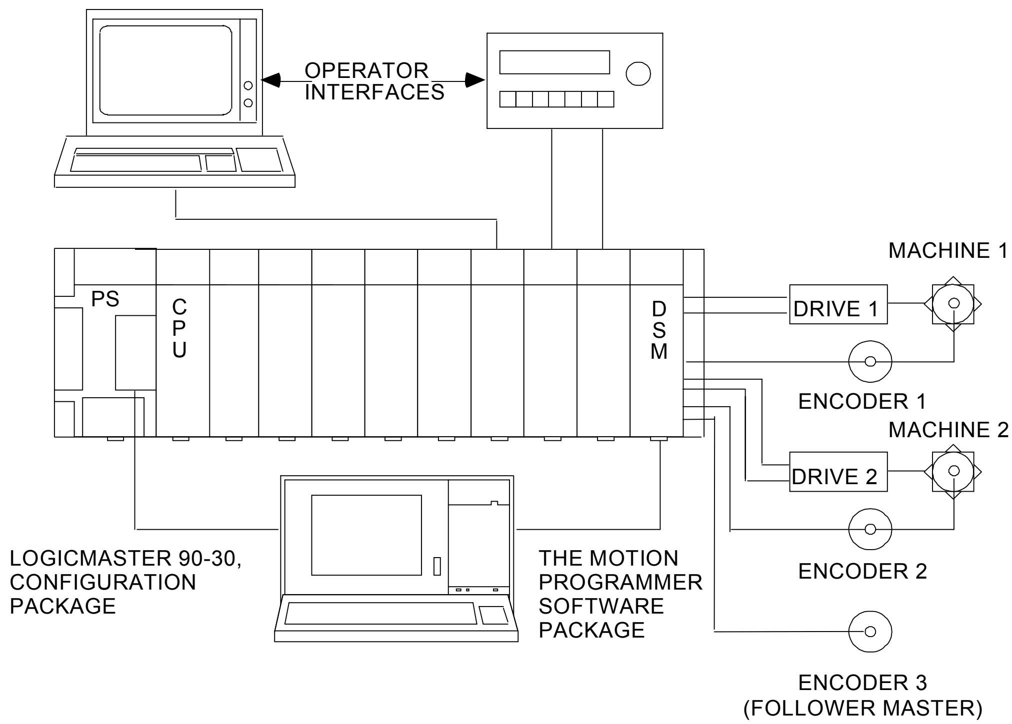


Figure 1-1. Hardware and Software Used to Configure, Program, and Operate a DSM302 Servo System

The Series 90-30 PLC and the DSM302

The DSM302 and Series 90-30 PLC operate together as one integrated motion control package. The DSM302 communicates with the PLC through the backplane interface. Every PLC sweep, data such as *Commanded Velocity* and *Actual Position* within the DSM302 is transferred to the PLC in %I and %AI data. Also every sweep, %Q and %AQ data is transferred from the PLC to the DSM302. The %Q and %AQ data is used to control the DSM302. %Q bits perform functions such as initiating motion, aborting motion, and clearing strobe flags. %AQ commands perform functions such as initializing position and loading parameter registers.

Besides the use of %I, %AI, %Q, and %AQ addresses, an additional way to send parameters to the DSM302 is with COMM_REQ instruction. Details about using the COMM_REQ instruction with the DSM can be found in Appendix B, *DSM Parameter Download Using the COMM_REQ*.

PLC Data Latency and DSM302 Latencies

The DSM302 is an intelligent module operating asynchronously to the Series 90-30 CPU module. Data is exchanged between the CPU and the DSM302 automatically. For information about the operation of the Series 90-30 sweep refer to the *Series 90-30 Programmable Controller Reference Manual GFK-0467. The following information specifies timing considerations as applied to the DSM302 module.*

- Motion program control or branching via faceplate inputs is a worst case 2 ms position loop execution delay *plus* the input filter delay (5 ms typical for 24 volt CTL inputs or 0.5 micro second input filter delay for 5 volt CTL inputs) of the input.

- PLC based functions may retrieve DSM status (%I and %AI) information from the DSM data memory asynchronously. The DSM will internally refresh all status data except *Actual Velocity* at a 2-millisecond data memory refresh rate. *Actual Velocity* is updated in the DSM data memory every 128 milliseconds. The DSM performs averaging to generate an accurate *Actual Velocity* reading and is therefore not intended for high-speed control purposes.

- The PLC requires 2-4 milliseconds back-plane overhead when reading and writing data to the DSM internal memory. The PLC will normally read input data from the DSM and write output data to the DSM once per PLC sweep. At worst case the PLC may have just missed the DSM 2 millisecond internal data update and will need to complete a sweep and begin another to read data from the DSM. The result is that DSM status data is available at a typical 4-6-millisecond rate or a PLC sweep, whichever is largest.

- The DSM302 may be configured for three different lengths of %AI and %AQ data. A PLC CPU requires time to read and write the data across the backplane with the DSM302. The *Series 90-30 Programmable Controller Reference Manual, GFK-0467 version L,* when available, will document the PLC sweep impact by CPU model group when the different configurations of %AI and %AQ data are selected.

- PLC commands to the DSM (%Q, %AQ) are normally output to the DSM at the end of the PLC sweep within one PLC sweep delay. The DSM will process the command within 4-5 milliseconds after receipt.

DSM302 Servo Loop Update Times

When controlling a GE Fanuc digital AC servo, the DSM302 uses these loop update times:

Table 1-1. Digital Servo Loop Update Times

Motor Current / Torque Loop:	250 microseconds
Motor Velocity Loop:	1 millisecond
Motor Position Loop:	2 milliseconds

When controlling an Analog servo, the DSM302 uses this loop update time:

Table 1-x. Analog Servo Loop Update Times

Motor Position Loop:	2 milliseconds

DSM302 Position Strobes

Each Position Feedback Input (Quadrature Encoder or GE Fanuc Serial Encoder) to the DSM302 includes two Position Strobe inputs. A rising edge pulse on a Strobe input causes the encoder *Actual Position* to be captured with a 250 microsecond maximum delay and within 2 ms is updated in the DSM data memory associated *Strobe Position* %AI data. The *Strobe Position* data is also stored in a DSM Parameter Register that can be used as an operand for Motion Program PMOVE and CMOVE commands.

In Digital mode, these strobes are 5V single-ended/differential inputs (IN1-IN2).

In Analog mode, these strobes are only 5V single-ended (IO5-IO6). In Analog mode only, these strobe inputs are pulled up high (as seen in the PLC %I Strobe status bits) if not physically connected to a device.

Configuration Software Packages

Logicmaster 90-30

The DSM302 is easily configured using the Logicmaster 90-30 Configuration software, version 8.02 or later. The DSM302 is assigned to a particular slot and rack like any other PLC module. In addition, other types of configuration data must be entered such as:

- I/O addresses where the CPU to DSM302 data transfers take place
- Serial Port Setup for connecting the Motion Programmer software
- DSM302 Setup Data
- A small optional motion program, Program Zero (20 lines in standard mode, 9 lines in follower mode)

All of the DSM302 configuration data is discussed in detail in Chapter 4, "Configuring the DSM302."

Windows Programming Software

GE Fanuc Control, the Windows-based PLC configuration software, will support the DSM302 module in Release 2.11 ServicePak 3 or later. The VersaPro releases will support the DSM302 also.

PC Control Programming Software

The PC Control configuration software will support the DSM302 in the later releases.

Motion Programming Software

Motion programs are normally created using one of the software packages described below, but a single short program (Program Zero) can be created within the Logicmaster 90-30 configuration software. Refer to Chapter 4, "Configuring the DSM302," for more information on Program 0.

Note

The DSM302 has a configurable serial port to communicate with the Motion Programmer. A special serial cable, IC693CBL316, is required to connect the DSM302 module to a PC for use with any of the supported motion programming software packages.

APM Motion Programming Software (MS-DOS)

The APM Motion Programmer operates much like the Logicmaster software package. It provides the capability of writing English language motion programs, storing the programs on disks, and downloading the programs to the DSM302 as desired. A Status Screen supports servo system monitoring. For detailed information on the Motion Programmer, refer to GFK-0664, *Series 90™ PLC Axis Positioning Module (APM) Programmer's Manual*.

Note

The APM Motion Programmer software was originally developed to program the APM300 products. Even though the APM Motion Programmer documentation does not reference the DSM302, it can still be used to program the DSM302. Version 1.51 or higher should be used to avoid problems with installations to large hard disks. It must be installed in a DOS environment. Refer to Appendix E, "Using the APM Motion Programmer With The DSM302," for specifics on using the APM Motion Programmer software to program and monitor the DSM302 module.

Operator Interfaces

Operator interfaces provide a way for the operator to control and monitor the servo system through a control panel or CRT display. These interfaces communicate with the PLC through discrete I/O modules or an intelligent serial communications or network communications module.

Operator data is automatically transferred between the PLC and the DSM302 through %I, %AI, %Q, and %AQ references which are specified when the module is configured. This automatic transfer of data provides a flexible and simple interface to a variety of operator interfaces that can interface to the Series 90-30 PLC.

Servo Drive and Machine Interfaces

The servo drive and machine interface is made through a 36-pin connector for each axis. This interface carries the signals that control axis position such as the Digital Pulse Width Modulated (PWM) signals to the amplifier, Digital Serial Encoder Feedback signals or Analog Servo Command and Quadrature Encoder Feedback. Also provided are *Home Switch* and *Axis Overtravel* inputs as well as general purpose PLC inputs and outputs.

Standard cables which connect directly to custom DIN rail or Panel mounted terminal blocks simplify user wiring, and are available from GE Fanuc. The terminal blocks provide screw terminal connection points for field wiring to the DSM302 module. Refer to chapter 3, "Installing and Wiring the Motion Mate DSM302" for more information concerning the cables and terminal blocks used with the DSM302 module.

Section 2: Overview of Product Operations

The DSM302 module may be operated in one of two modes:

Standard Mode

- In Digital Standard mode, the module provides closed loop position and velocity control for one or two servomotors.
- In Analog Standard mode, the module provides closed loop position control only for one or two servomotors. Note that when used with analog servos, velocity control is performed in the analog servo amplifier, not in the DSM module.
- For both digital and analog applications, user programming units can be adjusted by configuring the ratio of User Units and Feedback Counts configuration parameters. Jog, Move at Velocity and Execute Motion Program commands allow Standard mode to be used in a wide variety of positioning applications.

Follower Mode

- In Digital Follower mode, the module provides closed loop position and velocity control for one or two servomotors.
- In Analog Follower mode, the module provides only closed loop position control for one or two servomotors. Note that when the DSM is used with analog servos, velocity control is performed in the analog servo amplifier, not in the DSM module.
- In both digital and analog applications, the module provides most of the same features as Standard mode with the exception of a configurable User Units to Counts ratio (the ratio is fixed at 1:1 in Follower mode).
- In addition, a Master Axis position input can be enabled. Each Follower axis tracks the Master Axis input at a programmable A: B ratio. Motion caused by Jog, Move at Velocity and Execute Motion Program commands can be combined with follower motion generated by the master axis.
- Follower options include:
  - Selection of internal or external Master Axis sources
  - Acceleration Ramp to smoothly accelerate a slave axis until its position and velocity lock to the slave/master A:B ratio
  - Winder mode (traverse or spool winder) for coil winding and material handling applications

Standard Mode Operation

Figure 1-2 is a simplified diagram of the ***Standard*** mode Position Loop. An internal motion Command Generator provides *Commanded Position* and *Commanded Velocity* to the Position Loop. The Position Loop subtracts *Actual Position* (Position Feedback) from *Commanded Position* to produce a *Position Error*. The *Position Error* value is multiplied by a Position Loop Gain constant to produce the Servo Velocity Command. To reduce *Position Error* while the servo is moving, *Commanded Velocity* from the Command Generator is summed as a Velocity Feedforward term into the Servo Velocity Command output.

The following items discussed above are included in the data reported by the DSM302 to the PLC:

Commanded Velocity- the instantaneous velocity generated by the DSM302's internal path generator.

Commanded Position- the instantaneous position generated by the DSM302's internal path generator.

Actual Velocity- the velocity of the axis indicated by the feedback.

Actual Position- the position of the axis indicated by the feedback.

Position Error- the difference between the *Commanded Position* and the *Actual Position*.

The DSM302 allows a ***Position Loop Time Constant*** (in milliseconds) and a ***Velocity Feedforward*** (in percent) to be programmed. The ***Position Loop Time Constant*** sets the Position Loop Gain and determines the response speed of the closed Position Loop. The ***Velocity Feedforward*** percentage determines the amount of *Commanded Velocity* that is summed into the Servo Velocity Command.

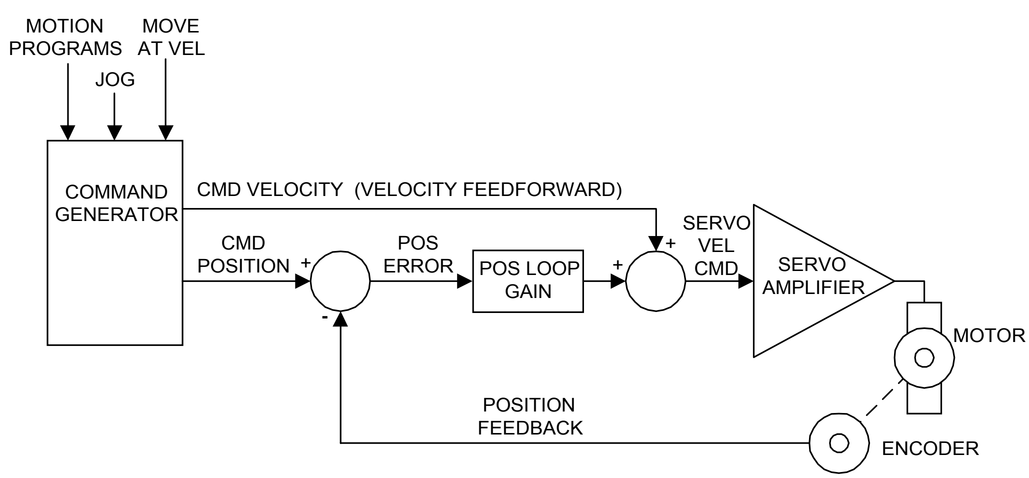


Figure 1-2. Simplified Standard Mode Position Loop with Velocity Feedforward

Follower Mode Operation

Figure 1-3 is a simplified diagram of the ***Follower*** mode Position Loop. It is similar to the ***Standard*** mode Position Loop (see previous page) with the addition of a Master Axis input. The Master Axis input is an additional command source producing a Master Axis Position and Master Axis Velocity. Master Axis Position is summed with *Commanded Position* from the axis Command Generator. Master Axis Velocity is summed with the *Commanded Velocity* (Velocity Feedforward) output of the axis Command Generator. **Therefore, the servomotor's position and velocity is determined by the sum of the Command Generator output and Master Axis input.** The Command Generator and Master Axis input can operate simultaneously or independently to create Servo Axis motion.

The DSM302 allows several sources for the Master Axis input:

- Aux Axis 3 Encoder (default Master input to Servo Axis 1 and 2)
- Internal Master Axis generator (selectable as Master input to Servo Axis 1 or 2 instead of Aux Axis 3 Encoder)
- Servo Axis 2 Encoder (configurable as Master input only to Servo Axis 1)
- Aux Axis 3 Analog Input 1 (configurable as Master Velocity input to Servo Axis 1 or 2)

The ratio at which a Servo Axis follows the Master Axis is programmable as the ratio of two integer numbers. For example, a Servo Axis can be programmed to move 125 Position Feedback units for every 25 Master Axis Position units. Each time the Master Axis Position changed by 1 position unit, the Servo Axis would move (125 / 25) = 5 Position Feedback units.

When Aux Axis 3 Analog Input 1 is used as the Master source, the Analog Input voltage is converted to a Master Axis Velocity and an equivalent Master Axis Position change. In this mode, the Servo Axis Velocity, not Servo Position, is proportional to the Analog Input value.

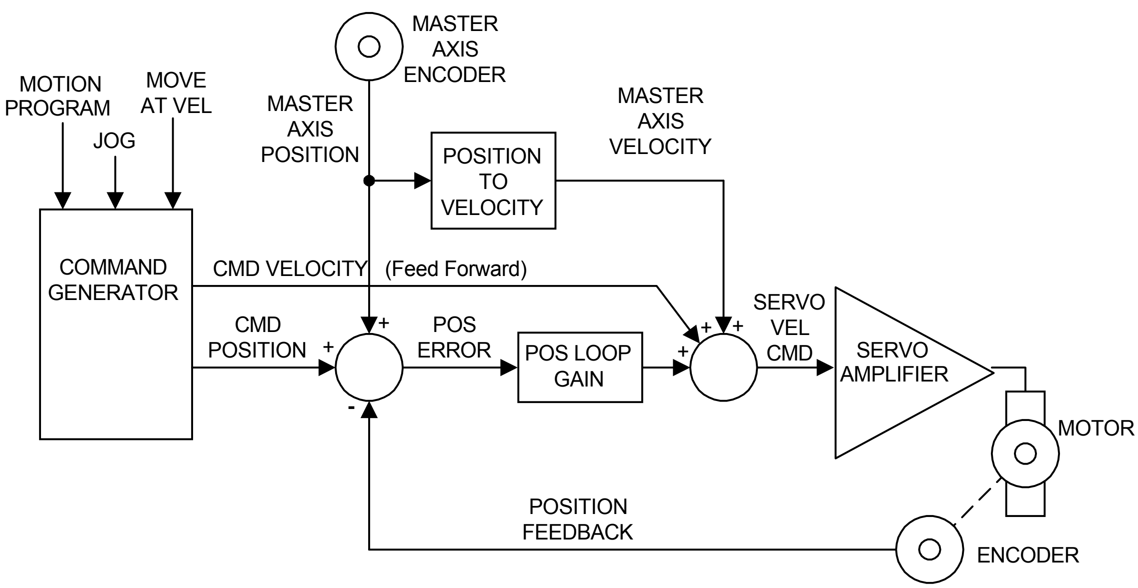


Figure 1-3. Simplified Follower Mode Position Loop with Master Axis Input

Section 3: α Series Servos (Digital Mode)

The GE Fanuc Digital α (pronounced “Alpha”) Series Servo features include:

- World leading reliability
- Low maintenance, no component *drift*, no commutator brushes
- All parameters digitally set, no retuning required
- Absolute encoder eliminates rehoming
- An optional brake is available
- Optional IP67 environmental rating is also available for most motors
- High resolution 64K count per revolution encoder feedback (incremental or absolute)

The GE Fanuc Servo motors, proven on over three million axes installed worldwide, offer the highest reliability and performance. The latest technologies such as high speed serial encoders and high efficiency Integrated Power Modules (IPM’s), further enhance customer benefits.

The GE Fanuc servo system is unique in that all the control loops - current, velocity and position - are closed in the motion controller. This approach reduces setup time and delivers significant throughput advantages even in the most challenging applications.

The servo drives are less costly to integrate and maintain. Control circuits are unaffected by temperature changes. There are no *personality* modules. The servos have a broad application range, that is, a wide load inertia range, flexible acc/dec and position feedback configurations, etc.

Extensive customization features are available to optimize performance and overcome machine limitations. IPM based servo amplifiers require 60% less panel space than conventionally switched amplifiers and produce 30% less heat.

α Series Integrated Digital Amplifier (SVU)

The α Series Integrated Servo Amplifiers (SVU) packages the amplifier with an integral power supply in a stand-alone unit. This unit is the same physical size and footprint as the previous “C” Series of GE Fanuc Servo Amplifiers.

The Integrated α Series SVU Amplifiers use the same connections as the “C” Series Amps except that the Emergency Stop circuit uses the internal 24v supply, thus there is no longer a requirement for a 100v power supply.

The heat sinks on the SVU design will mount through the panel to keep heat outside the enclosure.

Since the α SVU Amplifiers do not provide regeneration to line capability, discharge resistors may be required. These are available in several sizes.

SVU style α Series Servo Amplifiers are available in 5 sizes, with peak current limit ratings from 12 to 130 amps.

Cables to connect the SVU Amps to the DSM302 and to the motors are available in various lengths.

Refer to publication GFH-001, *Servo Product Specification Guide* for more information about the α Series servo products.

α Series FANUC Servo Motors

The α Series of servomotors incorporate design improvements to provide the best performance possible. Ratings up to 56 Nm are offered. These motors are up to 15% shorter and lighter than the previous *S* Series of servomotors. New insulation on the windings and an overall sealant coating help protect the motor from the environment.

The standard encoder supplied with the motor is a 64K absolute unit. Holding brakes (90 Vdc) and IP67 sealants are options. The α Series servomotors are approved to conform to international standards for CE (EMC and Low Voltage), IEC and UL/CUL. The following table indicates a sample of the α Series motors available (some αL, αC, αHV, and αM also available).

For more information refer to Chapter 4 of this manual, "Configuring the DSM302," under the section labeled "Motor Type." See also, the following publications:

- GFH-001, *Servo Products Specification Guide*
- GFZ-65142E/02, *α Series AC Motor Descriptions Manual*

Table 1-2. Selected α Series Servo Motor Models

α Model Number	Torque Nm	Output KW	Max. Speed (RPM)
α1	1	0.3	3000
α2	2	0.4	2000
α2	2	0.5	3000
α3	3	0.9	3000
α6	6	1.0	2000
α6	6	1.4	3000
α12	12	2.1	2000
α12	12	2.8	3000
α22	22	3.8	2000
α22	22	4.4	3000
α30	30	3.3	1200
α30	30	4.5	2000
α30	30	4.8	3000
α40	38	5.9	2000
α40/Fan	56	7.3	2000

Section 4: β Series Servos (Digital Mode)

The GE Fanuc Digital β (pronounced "Beta") Series Servo features include:

- World leading reliability
- Low maintenance, no component *drift*, no commutator brushes
- All parameters digitally set, no retuning required
- Absolute encoder eliminates rehoming
- Optional brake
- High resolution 32K count per revolution encoder

The GE Fanuc β Series Servos offer the highest reliability and performance. The latest technologies, such as high-speed serial encoders and high efficiency Integrated Power Modules, further enhance the performance of the servo system. Designed with the motion control market in mind, the β Series Servo Drives is ideally suited for the packaging, material handling, converting, and metal fabrication industries.

The GE Fanuc servo system is unique in that all the control loops - current, velocity, and position - are closed in the motion controller. This approach reduces setup time and delivers significant throughput advantages even in the most challenging applications.

The servo drives are less costly to integrate and maintain. Control circuits are unaffected by temperature changes. There are no *personality* modules. The servos have a broad application range including a wide load inertia range, flexible acceleration/deceleration and position feedback configurations. Extensive software customization features are available to optimize performance and overcome machine limitations.

β Series Digital Amplifiers

The β Series servo amplifier integrates a power supply with the switching circuitry. Therefore, GE Fanuc is able to provide a compact amplifier that is 60% smaller than conventional models. In fact, the β amplifier has the same height and depth as a GE Fanuc Series 90-30 PLC module. This allows efficient panel layout when using the DSM302 motion controller.

The amplifier is designed to conform to international standards.

GE Fanuc offers three communication interfaces for the β Series amplifiers: pulse width modulated (PWM), Fanuc Servo Serial Bus (FSSB), and I/O Link Interface. Only the pulse width modulated (PWM) interface may be used with the DSM302 module. The PWM interface utilizes the standard GE Fanuc servo communication protocol. Position feedback is communicated serially between the DSM controller and the motor mounted serial encoder.

β Series FANUC Servo Motors

The β Series Servomotors are built on the superior technology of the α Series servos. They incorporate several design innovations that provide the best possible combination of high performance, low cost, and compact size. Ratings of 0.5 to 12 Nm are offered.

These motors are up to 15% shorter and lighter than comparable servos. New insulation on the windings and an overall sealant coating help protect the motor from the environment.

The β Series motors conform to international standards (IEC). The motor protection level is IP65 (IP67 may be made available through special order).

A 32K absolute encoder is standard with each β Series servo. An optional 90 Vdc holding brake is also available with each model.

For more information refer to Chapter 4 of this manual, "Configuring the DSM302," under the section labeled "Motor Type." See also, the following publications:

- GFH-001, *Servo Products Specification Guide*
- GFZ-65232E, *β Series AC Motor Descriptions Manual*

Table 1-3. Selected β Series Servo Motor Models

β Model Number	**Torque Nm[1]**	**Output KW**	**Max. Speed RPM**
β0.5	0.5	0.2	3000
β1	1	0.3	3000
β2	2	0.5	3000
β3	3	0.5	3000
β6	6	0.9	2000
αC12	12	1.4	2000

1 - Continuous, 100% Duty Cycle

Note: The αC12 motor is listed with the β motors due to similar attributes

Section 5: SL Series Servos (Analog Mode)

Starting with firmware release 1.40, support for analog servos was added to the DSM302. This allowed the DSM302 to be matched with the GE Fanuc SL Series of analog servos to produce complete, high performance analog servo systems.

For details on the GE Fanuc SL Series Servo amplifiers, motors, and accessories, please see the *SL Series Servo User's Manual,* GFK-1581.

Chapter 2 Getting Started

Objectives of this chapter:

- To help you become familiar with the components and cables used in a DSM servo system as well as show you how they connect together.
- To show you how to verify your motion system connections and functionality.
- To identify supporting documentation providing detailed information on this subject.

This chapter serves as an introduction for the new user to the Motion Mate DSM302 motion system. Following through the steps outlined here, you should be able to operate or Jog your motion controller and servo in a short time. A minimum level of familiarity with the Logicmaster™ 90 Series 90™-30 software is assumed. You should have progressed through Appendix A, Configuration Lesson and Appendix B, Programming Lesson in the *Logicmaster™ 90 Series 90™-30/20/Micro Programming Software User's Manual*, GFK-0466.

Caution

Do not couple the motor shaft to mechanical devices for this exercise. Additionally, it is important that the servomotor be fastened securely to a stationary surface.

A typical DSM302 motion system includes the DSM302 motion controller, a Series 90-30 Programmable Logic Controller (PLC), motor(s), servo amplifier(s), I/O, and the Human Machine Interface (HMI).

The DSM302 control system consists of two parts: the servo control and the machine control.

The **servo control** translates motion commands into signals that are sent to the servo amplifier. The servo amplifier receives the control signals from the servo control and amplifies them to the required power level of the motor. The DSM302 provides the servo control.

The **machine control** (Series 90-30 PLC) houses the DSM302 module and I/O modules and executes user defined control logic. The machine control (PLC) and the servo control (DSM302) interface and exchange data over the Series 90-30 backplane.

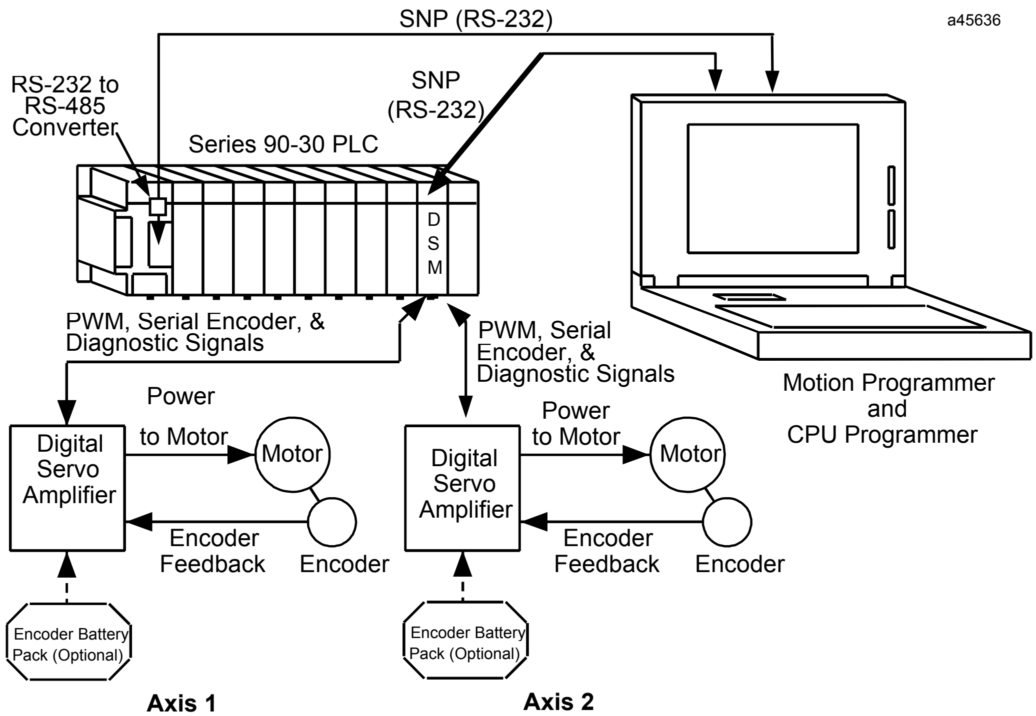


Figure 2-1. Typical 2-Axis Motion Mate DSM302 Digital Motion Control System

Section 1: Unpacking the System

The DSM302, Digital Servo Amplifiers, and Motors are packed separately. This section describes how to unpack the hardware and perform a preliminary check on the components.

Unpacking the DSM302

Carefully unpack the DSM302 and PLC system components. Verify that you have received all the items listed on the bill of material. Keep all documentation and shipping papers that accompanied the DSM302 motion system.

Unpacking the Digital Servo Amplifier

There are two digital amplifier and servo subsystem packages shipped for use with the DSM302, the α Series or the β Series.

The digital servo amplifier is shipped in a double-layered box. Remove the top layer of packing material to uncover the amplifier. Next, carefully remove the inner box from the outer layer. Then lift the amplifier out of the inner box. Retain any loose parts or gasket materials packed with the amplifier. Visually inspect the amplifier for damage during shipment.

Note

Do not attempt to change any pre-configured jumpers or switches on the amplifier at this time.

Unpacking the FANUC Motor

FANUC motors are packed two different ways, depending on their size. The largest motors are shipped on wooden pallets and are covered with cardboard. Most motors, however, are packed in cardboard boxes.

1. Unpacking Instructions:
   - For those FANUC motors packed in boxes, open the box from the top. The motors are packed in two pieces of form-fitted material. Carefully lift the top piece from the box. This should allow sufficient clearance for removing the motor.
   - If the motor is attached to a pallet, remove the cardboard covering. This will allow access to the bolts holding the motor to the pallet. Remove the bolts to free the motor from the pallet.
2. Inspect the motor for damage.
3. Confirm that the motor shaft turns by hand. NOTE: If the motor was ordered with the optional holding brake, the shaft will not turn until the brake is energized.

Let's start . . . .***Assembling the Motion Mate DSM302 System!***

Section 2: Assembling the Motion Mate DSM302 System

Before discussing specific assembly details, let's first review these general guidelines:

- Always make sure that the connectors lock into the sockets. The connectors are designed to fit only one way. Do not force them.
- Do not overlook the importance of properly grounding the DSM302 system components, **including the DSM302 faceplate shield ground wire**. Grounding information is included in this section.

All user connection, except for the grounding tab, are located on the front of the DSM302 module. The grounding tab is located on the bottom of the module. Refer to the figure below and take a few minutes now to familiarize yourself with these connections.

For instructions about installation of the DSM302 when IEC and other standards must be observed, see *Installation Requirements for Conformance to Standards*, GFK-1179.

Motion Mate DSM302 Connections

Figure 2-2 provides an overview of the faceplate and labels on the DSM302 module. For additional information and a complete connection diagram, please refer to chapter 3, *Installing and Wiring the Motion Mate DSM302*.

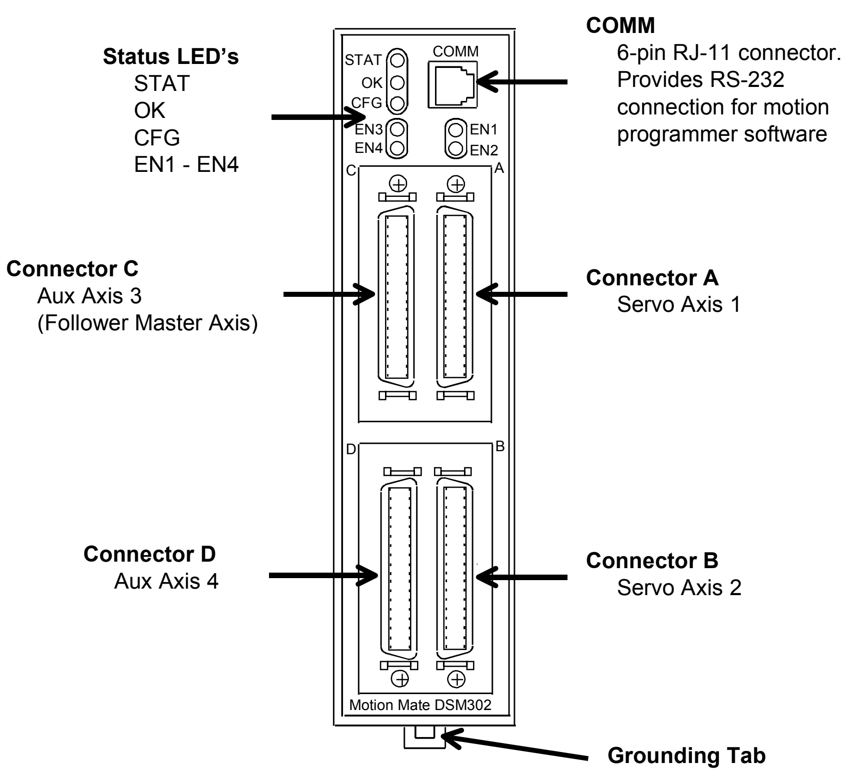


Figure 2-2. Face Plate Connections on the Motion Mate DSM302 Motion Control System

Connecting the α Series SVU Digital Servo Amplifier

Skip to the next section if you are connecting a β Series amplifier.

The α Series Digital Servo Amplifier does not require tuning adjustment during initial startup or when a component is replaced. It also does not need adjustment when environmental conditions change.

To connect the α Series Digital Servo Amplifier, follow the steps outlined below.

1. Connect the α Series Servo Amplifier to the DSM302.

A. Before connecting the servo command cable, make sure the DSM302 faceplate shield ground wire is connected. This wire is shipped with the DSM302 module and must be connected from the ¼ inch blade terminal on the bottom of the module to a suitable panel earth ground.

B. The servo command cable contains the pulse width modulated (PWM) output signal from the DSM, the serial data from the motor encoder, and diagnostic signals from the amplifier . The signals carried in this cable are at data communications voltage levels and should be routed away from other conductors, especially high current conductors.

C. Locate the servo command cable **IC800CBL001** (1 meter) or **IC800CBL002** (3 meter). Insert the mating end of this cable into the connector **JS1B**, located on the bottom of the Servo Amplifier (see Figure 2-4).

D. If you are not using the IC693ACC335 axis terminal board to break out user I/O such as overtravel or home limit inputs, insert the other end of the cable into the connector labeled **A**, for axis 1, or **B** for axis 2, on the front of the DSM302. If you are using the terminal board, insert the other end of the cable into the terminal block connector marked **SERVO**. Next locate the terminal board connection cable **IC693CBL324** (1 meter) or **IC693CBL325** (3 meter). Insert one end of this cable into the terminal board connector marked **DSM**. Insert the other end of the cable into the connector labeled **A**, for servo axis 1, or **B** for servo axis 2, on the front of the DSM302 module.

Note

Refer to “I/O Connections” in chapter 3 for information concerning the user I/O connections available on the IC693ACC335 terminal block.

SVU Amplifier Channel Switches

Confirm that the Channel Switches (dipswitches) located behind the SVU amplifier door are set as shown in the following tables. Note that the **OFF** position is to the left, and the **ON** position is to the right. Note also, that the switches are numbered from bottom to top (Switch 1 is the bottom switch). For example, in Figure 2-3, Switches 1, 3, and 4 are shown ON, and switch 2 is shown OFF

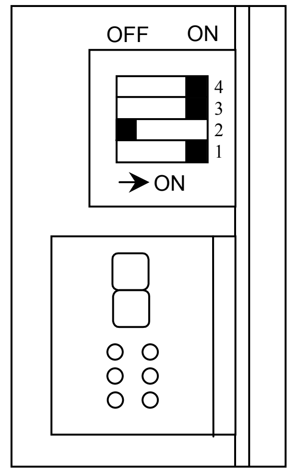


Figure 2-3. SVU Amplifier Channel Switches

Table 2-1. SVU Amplifier Channel Switch Settings

Amplifier SVU1-80				
Regenerative Discharge Unit	**SW1**	**SW2**	**SW3**	**SW4**
Built-in (100 W)	ON	OFF	ON	ON
Separate A06B-6089-H500 (200 W)	ON	OFF	ON	OFF
Separate A06B-6089-H713 (800 W)	ON	OFF	OFF	OFF

Amplifier SVU1-130				
Regenerative Discharge Unit	**SW1**	**SW2**	**SW3**	**SW4**
Built-in (400 W)	ON	OFF	ON	ON
Separate A06B-6089-H711 (800 W)	ON	OFF	ON	OFF

(To connect additional amplifiers, repeat steps B, C and D above for each additional amplifier.)

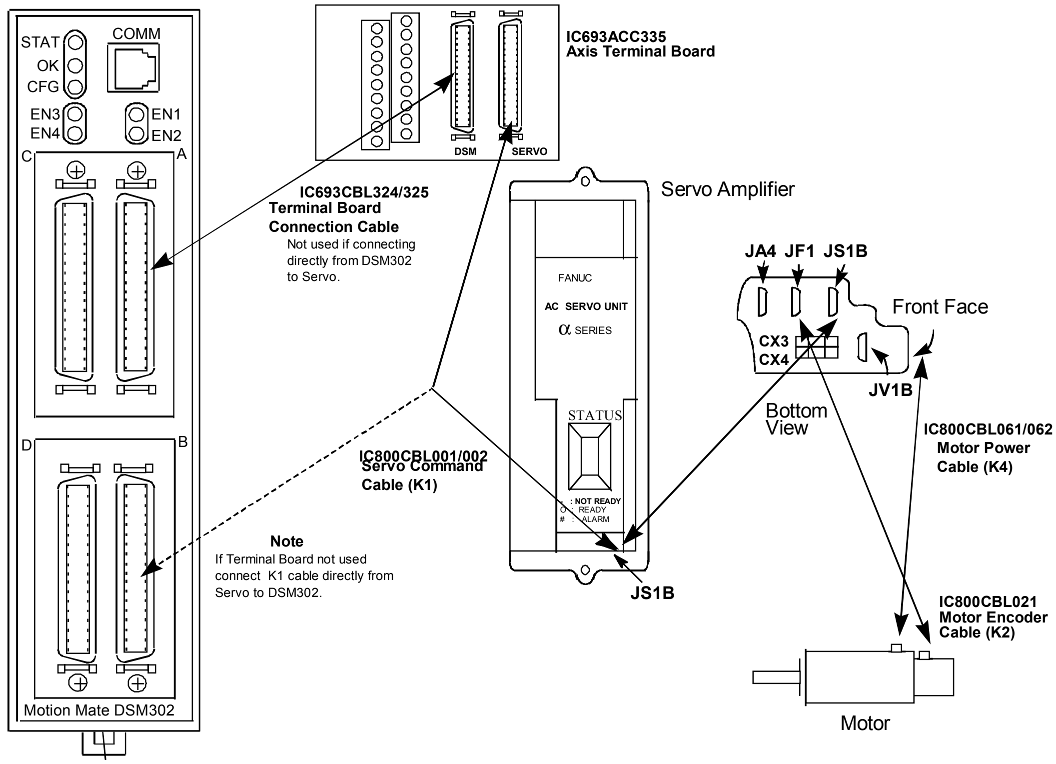


Figure 2-4. Connecting the α Series Digital Servo Amplifier to the Motion Mate DSM302

2. Connect the Motor Power Cable to the α Series Digital Servo Amplifier.

A. The size of the motor ordered for your system determines the K4 motor power cable you will use if you ordered prefabricated cables with your system. The motors in the following table are grouped to use one of the prefabricated cables available through GE Fanuc. This is not a complete listing of all α Series servomotor power cables, however the ones most commonly specified are included. A complete listing can be found in the *α Series AC Servo Motor Descriptions Manual*, GFZ-65142E.

Table 2-2. Prefabricated α Servo Motor Power Cable (K4) Part Number Examples

Motor Type	Severe Duty Cable Catalog Number	Cable Description	Cable Length
α3/3000 α6/3000	IC800CBL061	Elbow MS Connector	14 Meters
α12/3000 α22/2000 α30/1200	IC800CBL062	Elbow MS Connector	14 Meters
α30/3000 α40/2000	IC800CBL063	Elbow MS Connector	14 Meters

B. One end of this cable has four wires labeled U, V, W, and GND, which connect to screw terminals 9-12 on the servo amplifier. Connect these four wires to the terminal strip as shown in Figure 2-5.

C. Attach the other end of the cable to the motor after first removing the plastic caps protecting the motor’s connector. Note that this cable is keyed and can only be properly attached to one of the motor’s connection points.

(Repeat this procedure as needed for any other axes in the system.)

For the most current information on the motor power cables or wiring custom motor power cables please refer to the latest version of the α *Series Servo Motor Description Manual*, GFZ-65142E.

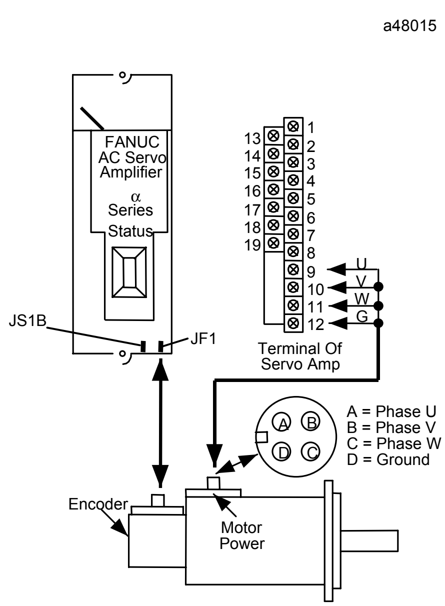


Figure 2-5. Connecting the Motor to the α Series Servo Amplifier Terminal Strip

3. Connect the Motor Encoder to the α Series Digital Servo Amplifier.

A. Remove the protective plastic cap from the encoder connector on the motor, and locate the K2 feedback cable **IC800CBL021**. The cable is configured so that it can only be attached to one connection on the motor.

B. Plug the opposite end into the connection labeled **JF1** on the bottom of the α Series servo amplifier (see Figure 2-6).

(Repeat this procedure for all axes in the system.)

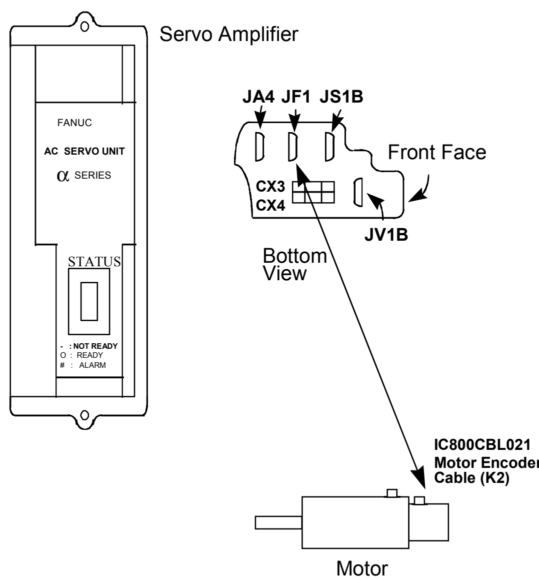


Figure 2-6. Connecting the α Series Motor Encoder

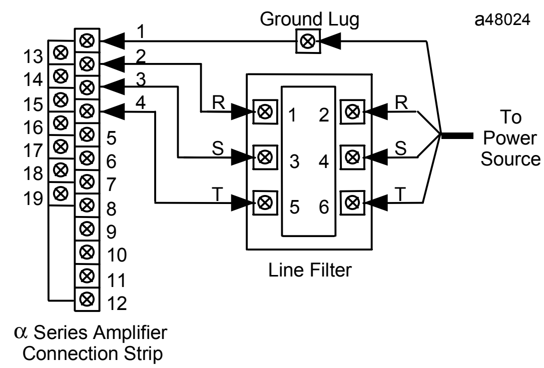
Table 2-3. Prefabricated α Servo Motor Encoder Cable (K2) for α3 to α40 Models

Motor Models	Severe Duty Cable	Cable Length
α3 to α40	IC800 CBL021	14 meters

Note: Details on α cables can be found in the *α Series AC Servo Motor Descriptions Manual*, GFZ-65142E, and in the *α and β Series Product Specifications Guide*, GFH-001.

4. Connect 220-Volt AC 3 Phase Power to the α Series Digital Amplifier

An AC line filter will reduce the effect of harmonic noise to the power supply; its use is recommended. Two or more amplifiers may be connected to one AC line filter if its power capacity has not been exceeded. Figure 2-6 shows how to connect the amplifier to the line filter.



NOTE: 220-Volt AC three phase power is required.

Figure 2-7. Connecting the Servo Amplifier to the Line Filter and Power Source

Note

You must supply the cable for both the connections between the line filter and the servo amplifier, and the connection between the line filter and the power source. Use 4-conductor, 600V, 60°C (140°F), UL or CSA approved cable between the line filter and the servo amplifier.

The gauge of wire used for connecting the line filter to the power source must be sized, based on the circuit breaker between the power source and the line filter and the number of servos connected to the line filter.

If a separate isolation transformer is used to supply AC power to the amplifiers, a line filter is not required.

5. Connect the Machine Emergency Stop to the α Series Digital Servo Amplifier

C. Pin 3 of connector **CX4, located on the bottom of the α Series (SVU) amplifier,** supplies +24 volts DC for the E-STOP circuit. Route this through the machine E-STOP circuit so that there is +24 volts DC to pin 2 when **not** in E-STOP. If no E-STOP switch is used this connection **must** be made with a wire jumper.

Note

You must supply the cable for this connection. Keyed connector plugs, marked as connector X and terminal connector pins are included with the

amplifier package. You must install this connection as a switch or jumper for the amplifier to operate.

Caution

Do not apply any external voltage to this connector.

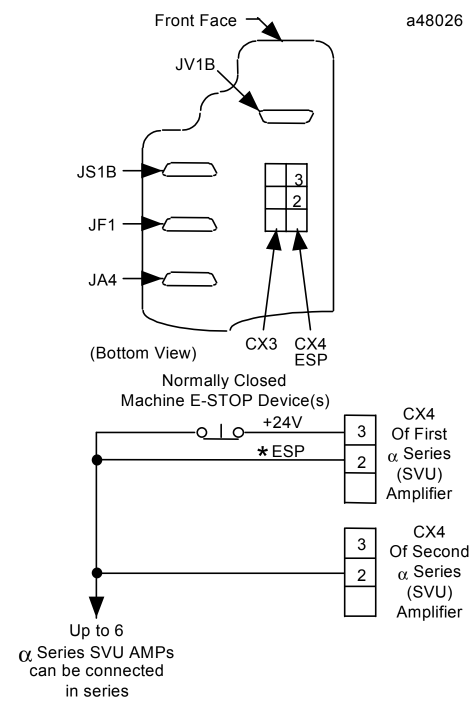


Figure 2-8. Connecting Emergency Stop to the α Series Servo Amplifier

For more information, refer to the *α Series Servo Amplifier (SVU) Descriptions Manual*, GFZ-65192EN.

Connecting the β Series SVU Digital Servo Amplifier

The β Series Digital Servo Amplifier does not contain any user adjustments. To connect the β Series Servo Amplifier, follow the steps outlined below. ***Refer to the previous section for α Series Amplifiers.***

1. Connect the β Series Digital Servo Amplifier to the DSM302

A. Before connecting the servo command cable, make sure the DSM302 faceplate shield ground wire is connected. This wire is shipped with the DSM302 module and must be connected from the ¼ inch blade terminal on the bottom of the module to a suitable panel earth ground.

B. The servo command cable contains the pulse width modulated (PWM) output signal of the motion controller, the serial data from the motor encoder, and diagnostic signals from the amplifier. The signals carried in this cable are at data communications voltage levels and should be routed away from other high current conductors.

C. Locate the servo command cable **IC800CBL001** (1 meter) or **IC800CBL002** (3 meter). Insert the mating end of this cable into the connector **JS1B**, located on the front of the Servo Amplifier (see Figure 2-9).

D. If you are not using the IC693ACC335 axis terminal board to break out user I/O such as overtravel or home limit inputs, insert the other end of the cable into the connector labeled **A**, for servo axis 1, or **B** for servo axis 2, on the front of the DSM302. If you are using the terminal board, insert the other end of the cable into the terminal board connector marked **SERVO**. Next locate the servo command cable **IC693CBL324** (1 meter) or **IC693CBL325** (3 meter). Insert one end of this cable into the terminal block connector marked **DSM**. Insert the other end of the cable into the connector labeled **A**, for servo axis 1, or **B** for servo axis 2, on the front of the DSM302.

(To connect additional amplifiers, repeat steps B - D above for each additional amplifier.)

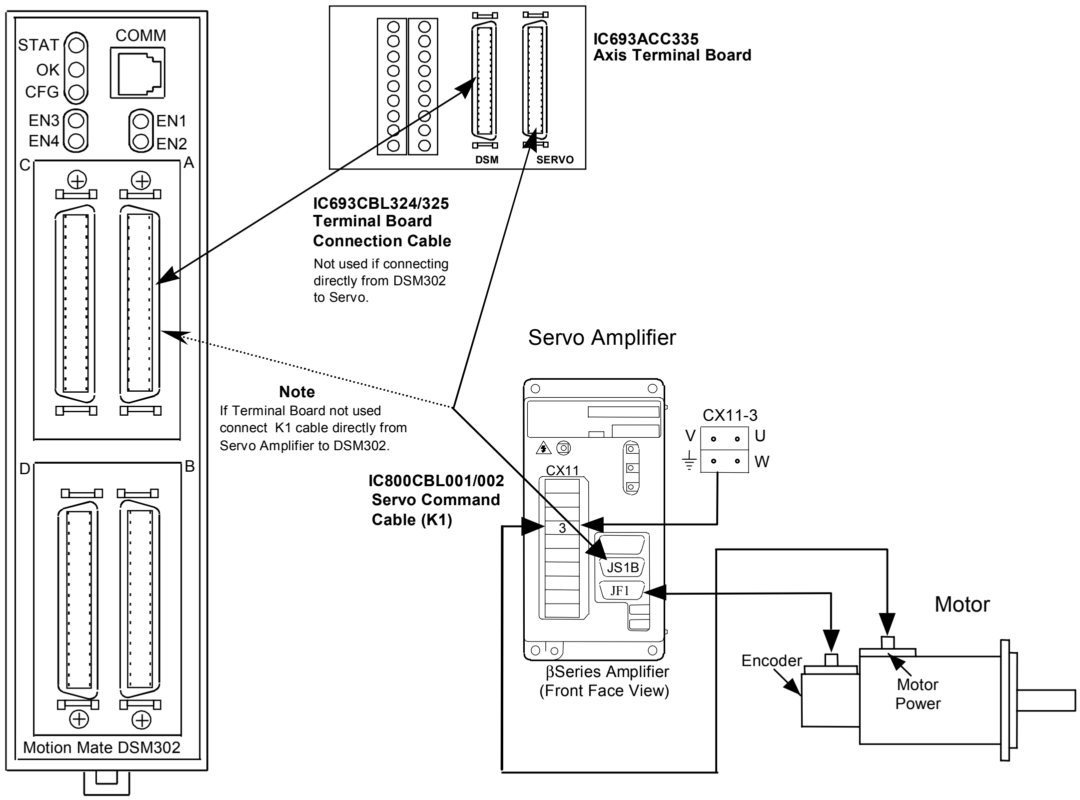


Figure 2-9. β Series Servo Amplifier Connections

For more information, refer to the connection section of the *Servo Product Specification Guide*, GFH-001.

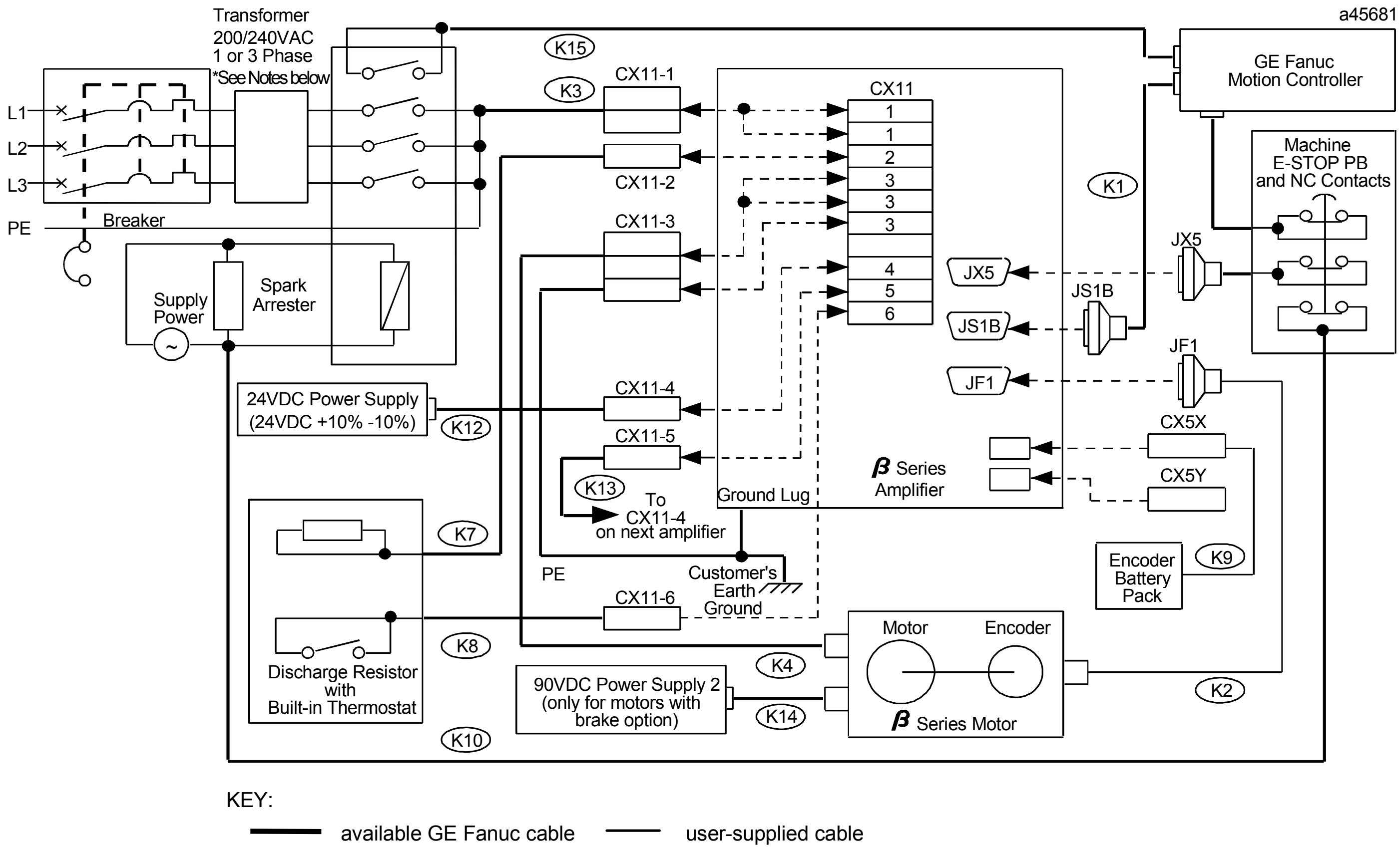
2. Connect the Motor Power Cable (K4) to the β Series Digital Servo Amplifier

Note: Make connections to the CX-11 connector carefully. This connector is not keyed. Double-check your connections before applying power. Incorrect connections could result in equipment malfunction or damage.

A. The size of the motor ordered for your system determines the motor power cable (K4) you must use. You can choose to purchase prefabricated cables or to build custom cables. Refer to the *β Series Control Motor Descriptions Manual*, GFZ-65232EN, for information about custom cables or installation for conformance to CE mark. The amplifier end of the prefabricated motor power cable is constructed to connect to terminal block **CX11-3** on the amplifier.

Table 2-4. K4 Cable –β Series Motor Cable Examples

Servo Motor Type	K-4 Motor Cable Part Number	Cable Description
β 0.5/3000	IC800CBL064	14 Meter Severe Duty
β 1/3000, β 2/3000, β3/3000, and β 6/2000	IC800CBL065	14 Meter Severe Duty
α C12/2000	IC800CBL066	14 Meter Severe Duty



*NOTES

1. Line filter and lightning surge absorber can be used in place of a transformer when 200-240 volts AC is available to the cabinet.
2. For single–phase operation, AC line phase L3 is not connected. Refer to the Servo System Specifications in the *Servo Product Specification Guide*, GFH-001 for output current de-rating.

Figure 2-10. Connecting the β Series Digital Servo Amplifier Terminal Strip

B. Attach the other end of the motor power cable to the motor, after first removing the plastic cap protecting the motor's connector. Note that this cable is keyed and can only be properly attached to one of the motor's connection points.

C. Motor power cables purchased from GE Fanuc will include a 1-meter, single conductor wire with a CX11-3 connector on one end and a ring terminal on the other. This cable provides grounding connections for the frame of the motor and should always be connected. Custom cable builders should always include this cable. See the previous connection diagram for proper connection to the amplifier.

(Repeat this procedure as needed for the other axis in the system.)

For more information, please refer to the *Servo Product Specification Guide*, GFH-001.

3. Connect the Motor Encoder Cable (K2) to the β Series Digital Servo Amplifier

The size of the motor ordered for your system determines the K4 motor power cable you will use if you ordered prefabricated cables with your system. Please refer to the table below to determine the correct encoder cable catalog number.

A. Remove the protective plastic cap from the connector on the motor, and locate the encoder cable K2, (see table 2-5). This cable has two distinct connectors.
B. Plug the end of the cable with the D-shell style connector into the connection labeled **JF1** on the servo amplifier (see Figure 2-9).
C. The other end of the cable is configured so that it can only be attached to one connection on the motor encoder (red end cap).

(Repeat this procedure for all axes in the system.)

Table 2-5. K2 Cable –β Series Encoder Cable Examples

Motor Type	K-2 Encoder Cable Part Number	Cable Description
β 0.5/3000	IC800CBL022	14 Meter Severe Duty
β1/3000, β 2/3000, β3/3000, and β6/2000	IC800CBL023	14 Meter Severe Duty
α C12/2000	IC800CBL021	14 Meter Severe Duty

4. Connect the 220 V AC Power Cable (K3) to the β Series Digital Amplifier

The AC power cable is a user-supplied cable, which connects to **CX11–1** on the face of the β Series amplifier. The connector for the amplifier end of this cable is part of kit A06B-6093-K301 supplied with each amplifier package. See the *Servo Product Specification Guide*, GFH-001, for more detailed information.

An AC line filter will reduce the effect of harmonic noise to the power supply; its use is recommended. A line filter is not needed if an isolation transformer or separate power transformer is used. Two or more amplifiers may be connected to one AC line filter or transformer as long as its power capacity is not exceeded. Figure 2-11 shows how to connect the amplifier to the line filter.

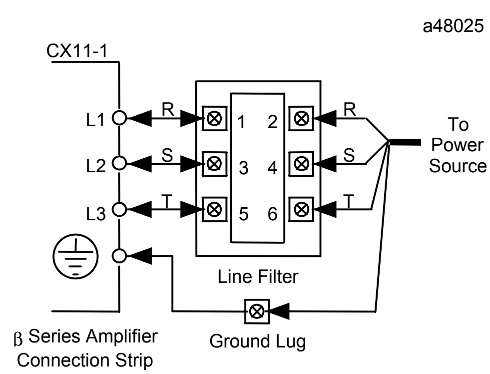


Figure 2-11. Connecting the β Series Servo Amplifier to the Line Filter and Power Source

Note

You must supply the cable for the connection between the line filter and the power source. Use 4-conductor, 600V, 60°C (140°F), UL or CSA approved cable between the line filter and the servo amplifier. The gauge of wire used for connecting the line filter to the power source must be sized, based on the size of the circuit breaker between the power source and the line filter and the number of servos connected to the line filter. The power connectors and terminals are supplied as part of the amplifier package.

5. Connect the Machine Emergency Stop to the β Series Digital Servo Amplifier

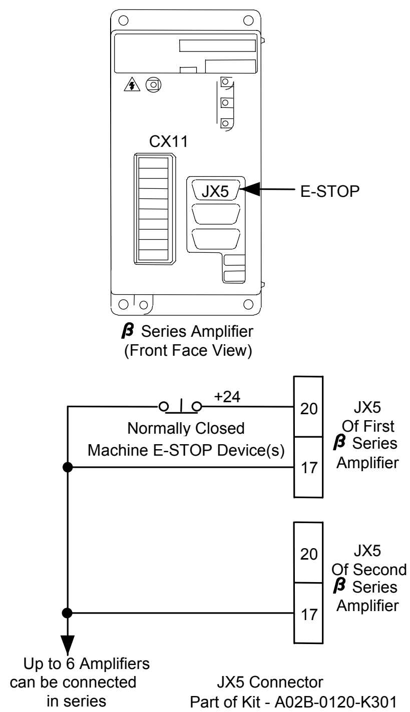


Figure 2-12. Connecting the E-STOP to the β Series Servo Amplifier

Note

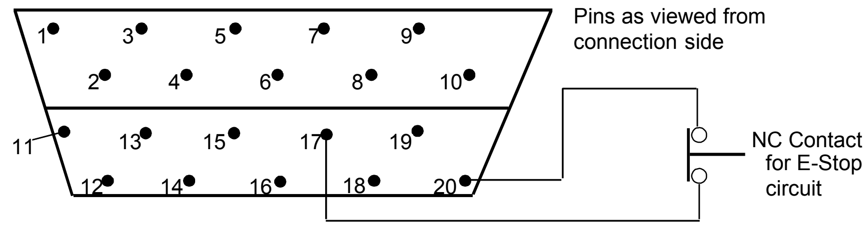
You must supply the cable for this connection package. The JX5 connector and connector cover is included with the amplifier as part number A02B-0120-K301. If no E-STOP circuit is required, this connection must be made with a wire jumper or the amplifier will not enable.

Connector **JX5 Pin 20** supplies +24V DC for the E-STOP circuit. Wire Pin 20 through a normally closed contact or switch so there is +24V DC to **JX5 Pin 17** when **not** in E-STOP. **GE Fanuc uses two brands of connectors for the JX5 connector. See figure 2-13 for proper connection to each type.**

CAUTION

Do not apply any external voltage to this connection.

HIROSE 20 Pin PCR Type Connector Pin Configuration



HONDA 20 Pin PCR Type Connector Pin Configuration

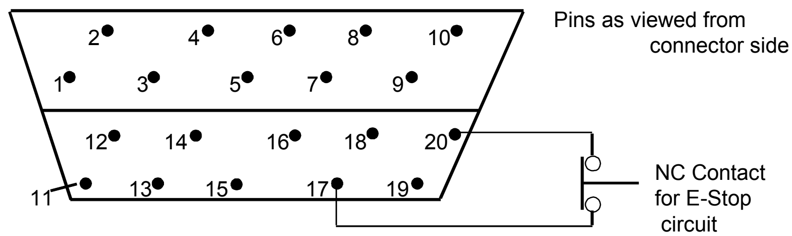


Figure 2-13. 20-Pin PCR Connector Pin-Out

6. Connect 24V DC Cable (K12) to the β Series Digital Servo Amplifier

A connector for the external 24 VDC supply is included with the amplifier package as a part of kit A06B-6093-K301 and should be connected to **CX11-4.** The other end of the cable must be connected to a 24VDC source capable of supplying at least 450 milliamps of current for each β Series amplifier. The GE Fanuc IC690PWR024 power supply is recommended. **Do not apply power at this time**.

7. Connect Cable K8 – Jumper or External Regeneration Resistor to the β Series Digital Servo Amplifier

Without External Regeneration Resistor (Using a Jumper)

If you do not have an external regeneration resistor, you must leave the connections on CX11-2 (DCP and DCC) open. However, you must jumper the CX11-6 (TH1 and TH2) terminals, shown in the figure below. (This jumper completes the circuit that would otherwise be completed by the normally closed thermal over-temperature switch in the external regeneration resistor unit.) If you do not have this jumper installed, the amplifier will not function. The jumper and its connector are included as a part of the connector kit, A06B-6093-K301, that is shipped with each amplifier package.

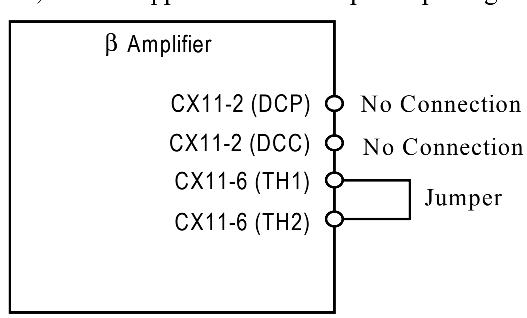


Figure 2-14. Installing a Jumper When an External Regeneration Resistor is Not Used

With External Regeneration Resistor

If you have an external regeneration resistor, observe that it has 4 wires. The two smaller wires (K8) connect to the resistor's internal, normally closed, over-temperature switch. This switch will open and shut down the amplifier if the resistor get too hot. The two larger wires (K7) connect to the resistor. All connectors needed to connect this resistor unit to the amplifier are provided in the amplifier package.

Connect the two over-temperature switch wires (K8) to CX11-6 terminals TH1 and TH2. (These connections are not polarity-sensitive.)

Connect the two resistor wires (K7) to CX11-2, terminals DCP and DCC. (These connections are not polarity-sensitive.)

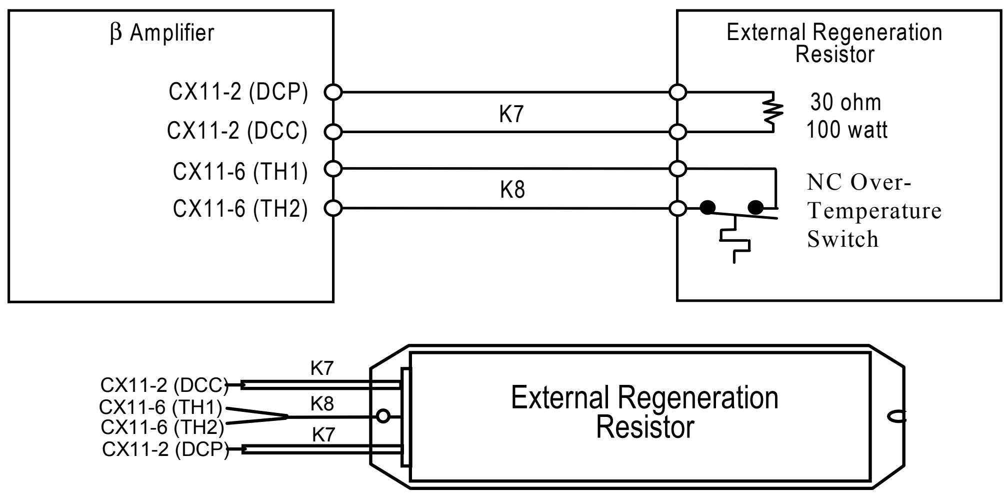


Figure 2-15. Connecting the External Regeneration Resistor

Installing and Wiring the DSM302 for Analog Mode

Important Analog Servo Considerations:

1. The Analog Servo Velocity Command output is a single-ended signal on pin 6 of the Auxiliary Terminal Board. This signal is referenced to 0v of the DSM module and PLC. This signal should be connected to the differential velocity command input of the servo amplifier.
2. The DSM302 provides a low current (30 ma) solid state relay output on pin 15 of theAuxiliary Terminal Board for connection to a servo amplifier enable input.
3. In analog mode, the DSM302 requires a Servo Ready input (IN_4 on wiring diagrams) on pin 5 of the Auxiliary Terminal Board. **This signal must be switched to 0v when the amplifier is ready to control the servo. If the servo amplifier does not provide a suitable output, the IN_4 input to the DSM302 must be connected to 0v.**
4. Quadrature encoder feedback in used in analog mode. Encoder wiring connections are detailed in figures 3-19 and 3-20.
5. Figures 3-19 and 3-20 are generic analog wiring diagrams for the DSM.
6. For details about the GE Fanuc SL Servo products, refer to the manual, *SL Series Servo User's Manual*, GFK-1581.

Grounding the Motion Mate DSM302 Motion System

The DSM302 System must be properly grounded. Many problems occur simply because this practice is not followed. To properly ground your Motion Mate DSM302 system, you should follow these guidelines:

- The grounding resistance of the system ground should be 100 ohms or less (class 3 grounding).
- The DSM302 faceplate shield ground wire (shipped with the module) must be connected from the ¼ inch blade terminal at the bottom of the module to a panel frame ground.
- If an axis terminal board is used, two shield (“S”) connections are provided and one of these must be connected to a panel frame ground.
- The system ground cable must have sufficient cross-sectional area to safely carry the accidental current flow into the system ground when an accident such as a short circuit occurs. Typically, it must have at minimum the cross-sectional area of the AC power cable. Figure 2-15 illustrates the grounding systems.
- The amplifier ground connections, power earth (PE) connections, and motor frame ground connections should always be wired to conform to local electrical wiring regulations. When installing in conformance to CE Mark directives, a grounding bar and clamp(s) (ordered separately) is required for the terminal block to amplifier cable. Refer to Chapter 3, Installing and Wiring the DSM302, I/O Cable Grounding section, for more details.

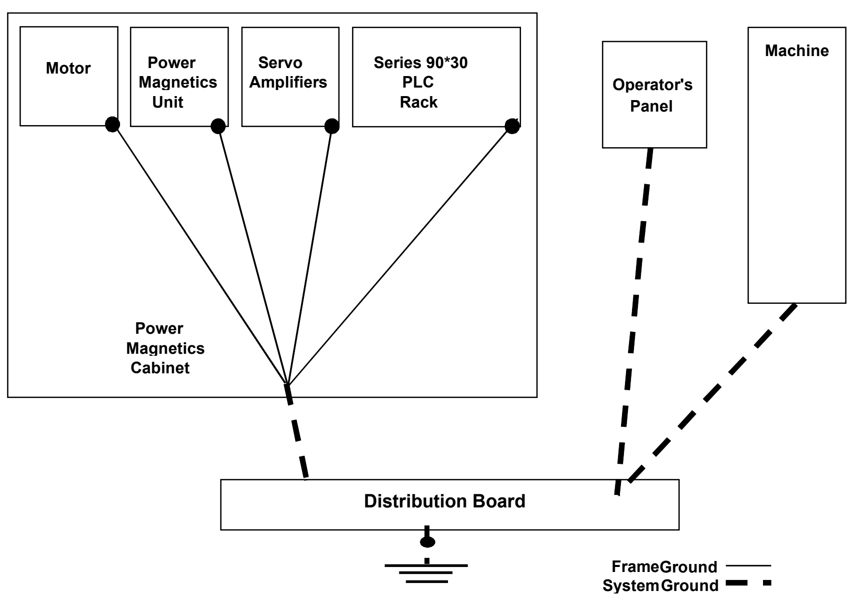


Figure 2-16. Motion Mate DSM302 System Grounding Connections

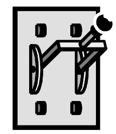
Table 2-6. Grounding Systems

Grounding System	Description
Frame Ground System	The frame ground system is used for safety and to suppress external and internal noises. In a frame ground system, the frames, cases of the units, panels, and shields for the interface cables between the units are connected.
System Ground System	The system ground system is used to connect the frame ground systems connected between devices or units with the ground.

This completes the steps required to assemble the Motion Mate DSM302 system.

You will have more work to do when interfacing to your machine, but for now, you can move on to the next step . . .

☞. . . . .***Turning on the Motion Mate DSM302 System!***



Section 3: Turning on the Motion Mate DSM302

Before turning on the power, you should:

- Confirm that the supplied cables are properly attached to the appropriate connectors.
- Confirm that all wiring to the power sources is correct.
- Make sure that the motors are properly secured.
- Check that all components are properly grounded, including the DSM302 faceplate shield.
- If you are using more than one motor, confirm that the servo amplifier connections and the feedback cables are not crossed between motors.

There is a specific sequence for turning on power to the DSM302 Control System. **In the order listed,** perform these steps:

1. Turn on the 220V AC power to the Digital servos. Verify that the *charged* LED indicator on the amplifier is on.
2. For β Series Digital amplifiers turn on the 24V DC source. Verify that the amplifier *Power* indicator is on.
3. Switch on the power to the Series 90-30 PLC, and check that the PWR LED on the Series 90-30 Power Supply is illuminated. Place the PLC in the **STOP/Disabled** mode. This can be done by using the Logicmaster 90-30 configuration software (Alt+R) hot key or with the Series 90-30 Hand Held Programmer (Mode Run –) keys.
4. If using an optional motor mounted holding brake, apply applicable power (90 VDC for α and β Series motors, and 24 VDC for SL Series motors) to brake leads to disengage the holding brake.

Having accomplished these steps, it's time to . . .

☞. . . . .***Configure the Motion Mate DSM302!***

Section 4: Configuring the Motion Mate DSM302

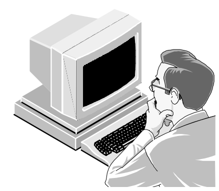
The DSM302 Controller is configured using the Logicmaster 90-30 configuration software version 8.02 or later.

The DSM302 has an extensive set of features that enable it to adapt to many different applications. You can easily make adjustments to your motion system. Parameter registers in the DSM302 memory allow you to use variables in DSM302 motion programs.

The DSM302 contains several configuration parameters; however, only a few need to be set for most applications. The remaining configuration parameters are normally set to their default values.

This section briefly describes the DSM302 configuration fields, and how to view and set the configuration parameters required to jog an axis using Logicmaster 90-30 software.

Using Logicmaster 90 (LM90) Program Folders



1. A Logicmaster 90-30 program folder is a sub-directory on your hard drive or floppy disk. The PLC configuration and logic program and the DSM302 configuration will all be stored in the LM90 folder.
2. Connect your personal computer to the PLC using the correct communication cable(s). The cable type will vary depending upon the communication method used by the Logicmaster 90-30 software (serial or Ethernet). For more information, please refer to GFK-0466, *Logicmaster 90 Series 90-30 Programming Software User's Manual* and GFK-0356, *Series 90-30 Programmable Controller Installation Manual.*
3. Start-up the Logicmaster 90-30 Programming Software, then select F2 for the configuration package. When prompted, select an existing working folder or create a new working folder by using the Folder Functions (Shift-F8) and then Select/Create (F1). This working folder will be used to hold a copy of the system configuration and ladder logic program.

   A. Select the working folder you have just created. Press I/O Configuration (Shift-F1, or F1 from the Configuration Package Main Menu). The main rack of the Series 90-30 PLC should now be displayed.

   B. Configure the components that make up your PLC system. The DSM302(s) are chosen to be placed in a particular rack/slot location(s) in the PLC. The configuration software must be used to put these modules in the same places as their physical locations. Select *Other* (F8) and *Motion* (F5). Then choose the DSM302 with the cursor key from the selections presented. Press the Enter key to make your choice and enter the configuration screens. *Escape* (ESC) can be used to save and exit when completed. Refer to the Logicmaster Programming Software User's Manual GFK-0466 for additional information.

Module Configuration Data

Using the Logicmaster 90-30 configuration package, with the rack configuration displayed:

1. Position the cursor on the DSM302 module and select the Zoom function key. Note that *page up* and *page down* select other screens of configuration data. The cursor keys move from field to field on the screen. The *tab* key toggles between alternate selections available in most non-numeric fields.

2. For now, select or enter the values from the following tables for the type of servo system, digital or analog, that you will be using. Later sections of this document provide additional detail for each configuration parameter, which will allow you to customize settings for your specific application requirements.

Table 2-7. Module Digital Mode Configuration Data Screen 1

Configuration Parameter	Description	Configure for Digital Mode	Configure for Analog Mode	Logicmaster Default Settings
Ref Adr	Start address for %I ref type (64 bits)	%I00001	%I00001	Next highest available %I reference
Ref Adr	Start address for %Q ref type (64 bits)	%Q00001	%Q00001	Next highest available %Q reference
Ref Adr	Start address for %AI ref type (40, 50 or 64 words)	%AI00001	%AI00001	Next highest available %AI reference
Ref Adr	Start address for %AQ ref type (6, 9 or 12 words)	%AQ00001	%AQ00001	Next highest available %AQ reference
AI/AQ Len	Number of PLC %AI and %AQ references used by the DSM module	40 / 6	40 / 6	40 / 6
Fdback Type	Feedback Type	**DIGITAL**	ENCODER	ENCODER
Ctl Loop	Control Loop Type	STANDARD	STANDARD	STANDARD
Servo Cmd	Servo Interface Type	**DIGITAL**	ANALOG	ANALOG
Motor1 Type Motor2 Type	GE Fanuc Motor Type	**From Table 2-8**	0	0
Motor1 Dir Motor2 Dir	Motor direction for positive velocity command	POS	POS	POS
Baud Rate	Baud rate of SNP Port	19200	19200	19200
Parity	Parity	ODD	ODD	ODD
Stop Bits	Number of stop bits	1	1	1
Data Bits	Number of data bits	8	8	8
Modem TT	Modem turn around time	0	0	0
Idle Time	Maximum link idle time	10	10	10
SNP ID	SNP ID	A00001	A00001	A00001

Motor Type (Digital Mode)

Selects the type of FANUC AC servomotor to be used with the DSM302. The DSM302 internally stores default setup motor parameter tables for each of the GE Fanuc servos supported. A particular motor for the indicated axis is selected via the Logicmaster 90-30 configuration fields *Motor1 Type* (for axis 1) or *Motor2 Type* (for axis 2). Supported motor types are listed in the table below.

FANUC Motor Model: Motor model information is in the form *series, continuous torque in Newton meters / maximum rpm.* Example: β2/3000 indicates a Beta (β) series motor of 2 Newton meters continuous torque capability and 3000 RPM maximum continuous speed.

FANUC Motor Specification: The configuration data for the motor type field is determined by the motor nameplate data. Motor part numbers are in the form A06B-xxxx-yyyy, where xxxx represents the Motor Specification field. For example: Reading the significant digits 0032 from a motor nameplate of A06B-0032-B078 indicates motor model β2/3000. The table below shows that 36 is the correct Motor Type Code for a Motor Specification number of 0032.

Table 2-8. FANUC Motor Type Code Selection

Motor Type Code	Motor Model	Motor Specification
61	α 1/3000	0371
46	α 2/2000	0372
62	α 2/3000	0373
15	α 3/3000	0123
16	α 6/2000	0127
17	α 6/3000	0128
18	α 12/2000	0142
19	α 12/3000	0143
27	α 22/1500	0146
20	α 22/2000	0147
21	α 22/3000	0148
28	α 30/1200	0151
22	α 30/2000	0152
23	α 30/3000	0153
30	α 40/2000	0157
29	α 40/FAN	0158
56	αL 3/3000	0561
57	αL 6/3000	0562
58	αL 9/3000	0564
59	αL 25/3000	0571
60	αL 50/2000	0572

Continued on next page

Table 2-8, Continued

Motor Type Code	Motor Model	Motor Specification
7	αC 3/2000	0121
8	αC 6/2000	0126
9	αC 12/2000	0141
10	αC 22/1500	0145
3	α 12HV/3000	0176
4	α 22HV/3000	0177
5	α 30HV/3000	0178
24	αM 3/3000	0161
25	αM 6/3000	0162
26	αM 9/3000	0163
13	β 0.5/3000	0113
35	β 1/3000	0031
36	β 2/3000	0032
33	β 3/3000	0033
34	β 6/2000	0034

Table 2-9. Additional Configuration Screen Information

Configuration Parameter	Description	Configure for Digital Mode	Configure for Analog Mode	Logicmaster Default Settings
User Units	User Units Value	1	1	1
Counts	Feedback Counts	1	1	1
OT Limit Sw	Over travel Limit Sw En/Dis	DISABLED	ENABLED	ENABLED
Pos EOT	Positive End of Travel	+8,388,607	+8,388,607	+8,388,607
Neg EOT	Negative End of Travel	– 8,388,608	– 8,388,608	– 8,388,608
Pos Err Lim	Position Error Limit	+ 40000	+ 40000	+ 4096
In Pos Zone	In Position Zone	10	10	10
Pos Loop TC	Position Loop Time Constant	60	1000*	1000
Vel at 10 V	Velocity for 10 V Output	+139820	4000*	4000
Vel FF %	Velocity Feed Forward	0	0	0
Intgr TC	Integrator Time Constant	0	0	0
Intgr Mode	Integrator Mode	OFF	OFF	OFF
Rev Comp	Reversal Compensation	0	0	0
DisDly	Drive Disable Delay	100	100	100
Vel Lp Gain	Velocity Loop Gain	16	16	16
Fdback Mode	Feedback Mode	INC	INC	INC
Jog Vel	Jog Velocity	+1000	+1000	+1000
Jog Acc	Jog Acceleration	+10000	+10000	+10000
Jog Acc Mod	Jog Acceleration Mode	LINEAR	LINEAR	LINEAR
Axis Mode	Axis Mode	CONTINU	CONTINU	CONTINU
Tuning Par1	Tuning Parameter 1	0	0	0
Tuning Dat1	Tuning Parameter 1's Data	0	0	0
Tuning Par2	Tuning Parameter 2	0	0	0
Tuning Dat2	Tuning Parameter 2's Data	0	0	0
Hi Limit	High Count Limit	+8,388,607	+8,388,607	+8,388,607
Lo Limit	Low Count Limit	–8,388,608	–8,388,608	–8,388,608
Home Positn	Home Position	0	0	0
Home Offset	Home Offset Value	0	0	0
Fnl Hm Vel	Final Home Velocity	+500	+500	+500
Find Hm Vel	Find Home Velocity	+2000	+2000	+2000
Home Mode	Find Home Mode	MOVE+	MOVE+	HOMESW

* Proper settings determined during system startup/tuning. See page D-14 in Appendix D.

Note

Repeat the above settings for axis 2 configuration if needed. We will not use Program 0 at this time so leave the Program 0 screens at their default values.

Store the Configuration to the PLC

You should complete the configuration of your Series 90-30 system to include the Power Supply, Rack, CPU and additional modules to match the target system. Consult the *Logicmaster 90 Series 90-30 Programming Software User's Manual* GFK-0466 as needed.

IMPORTANT

The completed configuration must be stored to the PLC. The Logicmaster configuration package menu selection F9 - UTILITY: Load/Store will allow you to complete this important step. Press Store (F2) and follow prompts to transfer the configuration to the PLC.

If all seems in order (that is, there are no PLC or IO status errors after Store, etc.), then it must be . . .**Motion Time**!

NOTE

A PLC status error of "System Configuration Mismatch" with the same rack/slot location as a DSM302 indicates that there is a parameter configured and sent to the DSM302 that has been rejected by the DSM302. Carefully check each parameter of your DSM302 configuration with the configuration settings in this manual for the discrepancy. Correct the discrepancy, clear the PLC Fault, and re-Store the configuration. Check that the error has been corrected. See the next section, Enabling Run Mode on the PLC, for instructions on viewing and clearing PLC faults.

Section 5: Testing Your System

Generating Motion

CAUTION

For correct machine operation, the recommended start-up procedure must be performed. This includes validating operation of the Overtravel Limits and Home Switch, checking for correct motor rotation direction, and tuning the velocity and position loops. This must be done by experienced personnel. For detailed start-up instructions, refer to Appendix D, *Start-Up and Tuning a GE Fanuc Digital or Analog Servo System.*

Enabling Run Mode on the PLC

The next step in the operation of the DSM302 system is to place the PLC in the RUN mode.

1. From the top level Logicmaster 90 Software menu select F1 . . . Logicmaster Programmer Package.
2. From the Programmer Package main menu select F3 Control and Status. Select F3 again for PLC Fault Table. Use the Alt + M hot key to place the control in the ONLINE mode (Status line at the bottom of the screen). Review the fault tables for any problems and then use the function key F9 - Clear to delete any PLC faults present. Select the F4 - I/O Fault Table and clear I/O faults as well.
3. Use the Alt + R hot key to place the PLC in RUN Mode.

Warning

Make sure that your motor is properly fastened down. If not, the motor could be damaged and/or cause injury to personnel.

4. Visually check that the PLC LED's PWR, OK and RUN are all on. Visually check that the DSM302 module LEDs STAT, OK and CFG are on. For α Series Digital amplifiers, visually check that the amplifier status readout is showing a minus sign (–) (which indicates Standby Mode). For β Series Digital amplifiers check that the Power LED is on and the ALM LED's are off. For SL Series amplifiers, check the front panel display on each amplifier for any errors.
5. If an amplifier indicates a fault, it may be necessary to completely remove power from the amplifiers and the PLC and repeat the power on sequence described in Section 3.
6. Assuming no errors are present, you should be able to jog the attached axes. For the startup configuration previously described, to clear any error indicated by the DSM302's STAT LED blinking, toggle (on then off) the *Clear Error %Q* bit (%Q0001) in the PLC data table. Conditions that continue to cause an error must be corrected in order to clear the status indicator. Refer to Appendix A, "Error Codes", for Error Status information.

Jogging With the Motion Mate DSM302

The ***Jog Velocity*, *Jog Acceleration***, and ***Jog Acceleration Mode*** are configurable in the DSM302 module. These values are used whenever a *Jog Plus* or *Jog Minus* %Q bit is turned ON. Note that if both Jog bits are ON, no motion is generated. Default jog values were set during configuration.

A *Jog* can be performed when no other motion is commanded. The *Enable Drive* %Q bit does not need to be ON to Jog. Turning on a *Jog* %Q bit will automatically turn on or enable the servo. You should hear the amplifier enable and the motor will have torque on the shaft. Conversely, the drive will automatically disable when the jog bit is released and completed.

To jog the axis in the positive direction, toggle ON %Q0021 for axis 1 or %Q0037 for axis 2. Note that the references used here are for %Q references starting at %Q0001. The starting %Q reference is configurable.

If the motor will not jog, skip over this section and go to Section 6, "Troubleshooting Your Motion System."

Section 6: Troubleshooting Your Motion System

Alarms

The first step in correcting a problem is to determine if any alarms have occurred. PLC alarms or errors may be viewed in the PLC fault table. Servo and motion subsystem alarms may be viewed in the DSM302 *Module Status Code* word, %AI0001; Servo *Axis 1 Error Code* word, %AI0002; and/or Servo *Axis 2 Error Code* word, %AI0003. Note that the references used here are for %AI references starting at %AI1. The starting %AI reference is configurable.

For more information on Motion Mate DSM302 alarms, please refer Appendix A, "Error Codes" which contains a list of alarm codes and descriptions. Check to see if this information can help you.

For more information on Troubleshooting, see Appendix D, Start-Up and Tuning a GE Fanuc Digital or Analog Servo System.

Configuration Settings

If your system powers up with alarms, it may be due to an incorrect configuration setting.

The Series 90-30 PLC configuration must be stored to the PLC CPU and the PLC must be in Run/Output Enabled mode.

If you cannot move an axis or execute a jog, check to see that all conditions necessary to perform these operations are met. Refer to the appropriate areas in this manual.

Getting Help

If you still cannot solve the problem, you may contact the GE Fanuc number for your area, listed in the table below:

Telephone Numbers

GE Fanuc Telephone Numbers	
Location	**Number**
North America, Canada, Mexico (Technical Support Hotline)	Toll Free: 1-800 GE Fanuc Direct Dial: (804) 978-6036
Latin America (for Mexico, see above)	Direct Dial: (804) 978-6036
France, Germany, Luxembourg, Switzerland, and United Kingdom	Toll Free: 00800 433 268 23
Italy	Toll Free: 16 77 80 596
Other European countries	352 (72) 79 79 309
Asia/Pacific – Singapore	65-566 4918
India	91-80-552 0107

GE Fanuc web site

http://www.gefanuc.com/support/plc/

Fax Link System

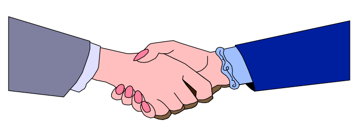
Help is also available in the form of Fax documents that you can order from the GE Fanuc Fax Link System. To access, please call (804) 978-5824. Follow the instructions for obtaining, by Fax, a master list of the available subjects. You may also download a master list (called Document 1) or print a set of Fax Link instructions from the following web address:

http://www.gefanuc.com/support/plc/fax.htm

When you receive the master list, select a document, then call and specify the document number and it will be Faxed to you.

Once you have successfully moved an axis, it's . . .

Mission Accomplished!



Section 7: Next Steps to Take

After successfully moving an axis, what's next?

This startup chapter does not cover all aspects of the DSM302 motion system. For this reason, you should review the information provided in all the manuals (see *Related Publications* in the Preface of this manual). Additionally, GE Fanuc offers applicable training courses. If your application requires a custom machine interface, you should also attend a GE Fanuc programming course. For information about training, call 1-800-GE FANUC or contact your GE Fanuc Sales representative.

The installation and wiring chapter of this manual will be your guide to completing the DSM hardware installation.

The configuration chapter will provide details needed to configure the DSM module for your application. You should begin by reviewing the *Configuration Parameters* section in chapter 4.

_ . . . . . . ***Two final recommendations****:*

- Save the paperwork that came with your system.

  The *Important Product Information* sheet will contain the latest information on this product, some of which may not be included in this manual.

- Back up your ladder logic folder.

  Important! . . . . Do this frequently while developing your application.

Chapter 3 Installing and Wiring the DSM302

Section 1: Hardware Description

This section identifies the module's major hardware features. The module's faceplate provides seven status LEDs, one communications port RJ-11 connector and four user I/O connectors (36 pin). A grounding tab on the bottom of the module provides a convenient way to connect the module's faceplate shield to a panel ground.

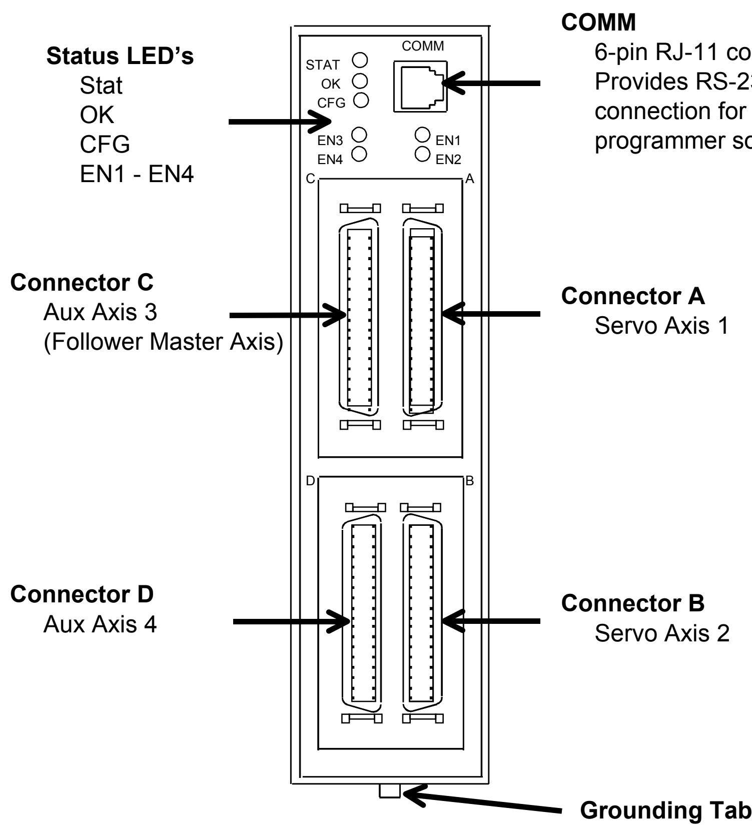


Figure 3-1. DSM302 Module

LED Indicators

There are seven LEDs on the DSM302 module which provide status indications. These LEDs are described below.

STAT Normally ON. FLASHES to provide an indication of operational errors. Flashes *slow* (four times/second) for Status-Only errors. Flashes *fast* (eight times/second) for errors which cause the servo to stop.

ON: When the LED is steady ON, the DSM302 is functioning properly. Normally, this LED should always be ON.

OFF: When the LED is OFF, the DSM302 is not functioning. This is the result of a hardware or software malfunction which will not allow the module to power up.

Flashing: When the LED is FLASHING, an error condition is being signaled.

Constant Rate, CFG LED ON:

The LED flashes slow (four times/second) for Status Only errors and fast (eight times/second) for errors which cause the servo to stop. The operational error code will be placed in one of the first four %AI status words and the *Module Error Present* %I status bit will be ON.

Constant Rate, CFG LED Flashing:

If the STAT and CFG LEDs both flash **together** at a constant rate, the DSM302 module is in boot mode waiting for a new firmware download. If the STAT and CFG LEDs both flash **alternately** at a constant rate, the DSM302 firmware has detected a software watchdog timeout due to a hardware or software malfunction.

Irregular Rate, CFG LED OFF:

If this occurs immediately at power-up, then a hardware or software malfunction has been detected. The module will blink the STAT LED to display two error numbers separated by a brief delay. The numbers are determined by counting the blinks in both sequences. Record the numbers and contact GE Fanuc for information on correcting the problem.

OK The OK LED indicates the current status of the DSM302 module.

ON: When the LED is steady ON, the DSM302 is functioning properly. Normally, this LED should always be ON.

OFF: When the LED is OFF, the DSM302 is not functioning. This is the result of a hardware or software malfunction which will not allow the module to power up.

CFG This LED is ON when a valid module configuration has been received from the PLC. Flashes *slow* (four times/second) during the Motion Program Store function. Flashes *fast* (eight times/second) during the Write User RAM to EEPROM operation.

EN1 When this LED is ON, the servo drive for Servo Axis 1 is enabled.

EN2 When this LED is ON, the servo drive for Servo Axis 2 is enabled.

EN3 When this LED is ON, the *Force Analog Output* command for Aux Axis 3 is active.

EN4 When this LED is ON, the *Force Analog Output* command for Aux Axis 4 is active.

The DSM COMM (Serial Communications) Connector

The module's front panel contains a single RJ-11 connector for serial communications, labeled "COMM". This port has two main uses:

- It is used to upload or download motion programs between the DSM302 and a computer running the GE Fanuc APM Motion Programmer software.
- It is used to download firmware updates to the DSM module from a personal computer running the GE Fanuc PC Loader utility software. See Appendix F for details.

The serial COMM port connects to the personal computer's serial port. It uses the GE Fanuc SNP protocol and is RS-232 compatible. The baud rate is configurable from 300 to 19,200 baud. The COMM port is configured using the Logicmaster 90-30 Configuration Software.

A 1-meter cable, IC693CBL316, is available from GE Fanuc to connect to a personal computer. This cable uses a 9-pin female D-shell connector for the computer side and an RJ-11 connector for the DSM302. **If a longer cable is used, the maximum recommended length is 50 feet.**

Table 3-1. DSM302 COMM Port Pin Assignments

RJ-11 Pin Number	9-Pin (female) Number	Signal Name	Description
1	7	CTS	Clear to Send
2	2	TXD	Transmit Data
3	5	0V	Signal Ground
4	5	0V	Signal Ground
5	3	RXD	Receive Data
6	8	RTS	Request to Send

(Pin 1 is at the bottom of the connector when viewed from the front of the module.)

I/O Connectors

The DSM302 is a two-axis servo controller with four 36-pin I/O connectors labeled A, B, C, and D. The connectors are assigned as follows:

Table 3-2. Axis I/O Connector Assignments

Connector	Axis Number	Axis Type	I/O Usage
A	1	Servo Axis	Closed Loop Digital or Analog Servo Control
B	2	Servo Axis	Closed Loop Digital or Analog Servo Control
C	3	Aux Axis	Position Feedback and auxiliary analog / digital I/O
D	4	Aux Axis	Auxiliary analog / digital I/O

Both the A and B connectors provide the same analog and digital I/O circuits. Connectors A and B are both used for either analog or digital servo control. (Note that if the DSM302 is used to

control two axes they both must be analog or they both must be digital – they cannot be one of each.) The module's C and D connectors are used for auxiliary functions such as Follower Master axis position feedback and additional analog / digital I/O.

Any of these four connectors used in a system typically is cabled to an appropriate Terminal Board with cable IC693CBL324 (1 meter) or IC693CBL325 (3 meters). There are three different types of terminal boards that provide screw terminals for connecting to external devices, described later in the "Terminal Board" section of this chapter.

Shield Ground Connection

The DSM302 faceplate shield must be connected to frame ground. This connection from the DSM302 to frame ground can be made using the green ground wire (part number 44A735970-001R01) provided with the module. This wire has a stab-on connector on one end for connection to a ¼ inch terminal on the bottom of the DSM302 module and a terminal on the other end for connection to a grounded enclosure.

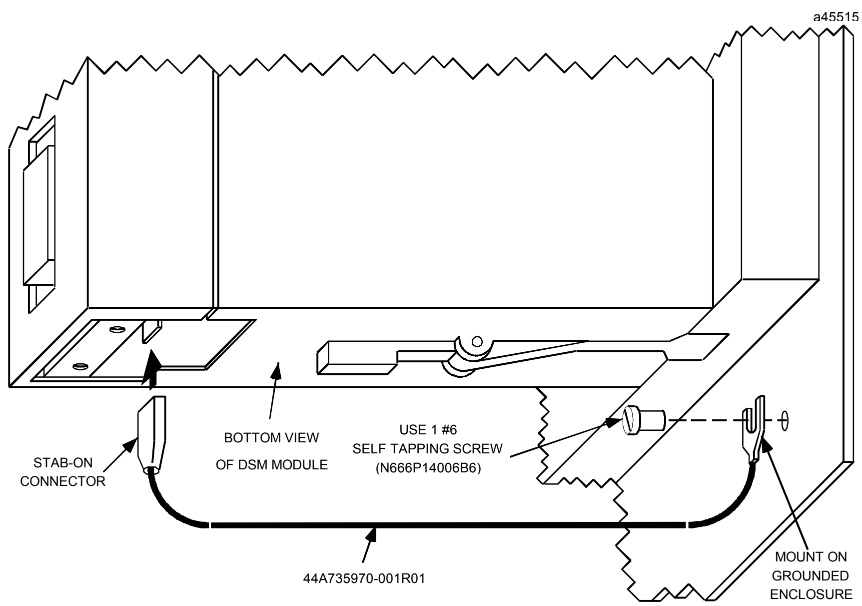


Figure 3-2. Connecting the Shield Ground

Section 2: Installing the DSM302 Module

The Motion Mate DSM302 can operate in *any* Series 90-30 CPU, expansion, or remote baseplate (Series 90-30 release 6.50 or later). The configuration files created by Logicmaster 90-30 Configuration software must match the physical configuration of the modules.

To install the DSM302 on the baseplate, follow these steps:

1. Use the Logicmaster 90-30 Configuration software or the Hand-Held Programmer to stop the PLC. This will prevent the local application program, if any, from initiating any command that may affect the module operation on subsequent power-up.
2. Power down the Series 90-30 PLC system.
3. Align the module with the desired base slot and connector. Tilt the module upward so that the top rear hook of the module engages the slot on the top edge of the baseplate.
4. Swing the module down until the connectors mate and the lock-lever on the bottom of the module snaps into place engaging the baseplate notch.
5. Connect the faceplate shield wire from the ¼ inch blade terminal on the bottom of the module to a suitable panel earth ground.
6. Refer to Figures 3-10 through 3-23 and Tables 3-7 through 3-14 for I/O wiring requirements.
7. Power up the PLC rack. The Status LED of the Motion Mate DSM302 will turn ON when the controller has passed its power-up diagnostics.
8. Repeat this procedure for each DSM302 module in your PLC system.
9. Configure the DSM302 module(s) as described in Chapter 4.

The following table lists the DSM302 module current draw and defines the number of modules that can be installed in a particular PLC system.

The number of modules in a system may be restricted by:

- PLC rack power supply capacity
- PLC I/O Table space. The DSM module requires the use of %I, %Q, %AI, and %AQ memory in the PLC's I/O Table, with the %AI type usually being the most restrictive of the four.
- PLC Configuration data storage capacity
- PLC Configuration software memory

The absolute limits for each PLC type must not be exceeded because in some cases they are based on PLC I/O Table and Configuration data capacity.

The practical number of axes must consider I/O use and sweep time of the entire system. The number of DSM302 modules might be additionally limited by the memory available in the personal computer running Logicmaster 90-30 configuration software.

Table 3-3. Maximum Number of DSM Modules per System

Power Supply Voltage: **Power Supply Current Draw by DSM:**	5 VDC from PLC backplane 800 mA plus encoder supply current (see next item).
Available +5V Current/Module to supply external encoder, if used:	500 mA (if used, must be added to module +5v current draw)
Maximum Number of modules/system:	
Model 311, 313, 321, 323 PLCs : **(5 or 10-slot CPU baseplates)**	1 DSM302 module in CPU baseplate
Model 331 PLC : **(5 and 10-slot CPU baseplates, 5 and 10-slot expansion or remote baseplates - 5 total baseplates per system)**	2 DSM302 modules in CPU baseplate with PWR321/322 3 DSM302 modules in CPU baseplate with PWR330/331 3 DSM302 modules in expansion/remote baseplate ***3 total DSM302 modules per PLC system***
Model 340, 341 PLC : **(5 and 10-slot CPU baseplates, 5 and 10-slot expansion or remote baseplates - 5 total baseplates per system)**	2 DSM302 modules in CPU baseplate with PWR321/322 6 *DSM302 modules in CPU baseplate with PWR330/331* 3 DSM302 modules in expansion/remote baseplate with PWR321/322 *7 DSM302 modules in expansion/remote baseplate with PWR330/331* ***8 total DSM302 modules per PLC system with PWR321/322*** ***8 total DSM302 modules per PLC system with PWR330/331***
Model 350 - 364 PLC: **(5 and 10-slot CPU baseplates, 5 and 10-slot expansion or remote baseplates - 8 total baseplates per system)**	2 DSM302 modules in CPU baseplate with PWR321/322 *6 DSM302 modules in CPU baseplate with PW330/331* 3 DSM302 modules in expansion/remote baseplate with PWR321/322 *7 DSM302 modules in expansion/remote baseplate with PWR330/331* ***23 total DSM302 modules per PLC system with PWR321/322*** ***32 total DSM302 modules per PLC system with PWR330/331****

The total power consumption of all modules in a baseplate must be calculated so that the total load capacity of the supply is not exceeded. Refer to GFK-0356, *Series 90-30 PLC Installation Manual,* for detailed information on load requirements for Series 90-30 modules. The available power supplies are:

- *IC693PWR321* - Standard AC/DC Power Supply - allows 15 watts (3000 ma) for +5 VDC
- *IC693PWR322* - 24/48 VDC Power Supply - allows 15 watts (3000 ma) for +5 VDC
- *IC693PWR330* - High Capacity AC/DC Power Supply - allows 30 watts (6000 ma) for +5 VDC
- *IC693PWR331* - High Capacity 24 VDC Power Supply - allows 30 watts (6000 ma) for +5 VDC

Note

Refer to GFK-0867B, (GE Fanuc Product Agency Approvals, Standards, General Specifications), or later version for product standards and general specifications.

Note

Installation instructions in this manual are provided for installations that do not require special procedures for noisy or hazardous environments. For installations that must conform to more stringent requirements (such as CE Mark), see GFK-1179, *Installation Requirements for Conformance to Standards.*

Section 3: I/O Wiring and Connections

I/O Circuit Types

Each of the module's four connectors (Connector A, B, C, and D) provide the following types of I/O circuits:

- Three differential / single ended 5v inputs (IN1-IN3)
- 5 VDC Encoder Power (P5V)
- One single ended 5v input (IN4)
- Four single ended 5v input / output circuits (IO5-IO8)
- Three 24v inputs (IN9-IN11)
- One 24v, 125 ma solid state relay output (OUT1)
- Two differential 5v line driver outputs (OUT2-OUT3)
- One 24v, 30 ma solid state relay output (OUT4)
- Two differential +/- 10v Analog Inputs (AIN1-AIN2)
- One single ended +/- 10v Analog Output (AOUT1)

Not all of these I/O circuits are available for user connections. Some of the circuits are used to control the GE Fanuc digital servo amplifier. Refer to Tables 3-11 through 3-14 for additional information.

Terminal Boards

- **Axis Terminal Board, Catalog No. IC693ACC335** - Used in Digital mode only. It connects DSM connector A or B to a GE Fanuc α or β Digital Servo amplifier. It also provides screw terminal connections for I/O devices. This terminal board contains two 36 pin connectors. One connects to the DSM via cable IC693CBL324/325, and the other connects to the GE Fanuc Digital Servo amplifier via the servo command cables IC800CBL001 / 002. See Figures 3-10, 3-16, 3-17, and 3-18.

  Note: For Digital Servo applications that do not require use of the DSM's A or B connector I/O signals, the DSM connector can be cabled directly to the Digital servo amplifier. Refer to Section 3, "I/O Wiring and Connections," later in this Chapter for additional information.

- **Auxiliary Terminal Board, Catalog No**. **IC693ACC336 –** This terminal board contains a single 36 pin connector which connects to the DSM302 module. This board has two basic applications (see Figures 3-10 and 3-11):

1. In Analog mode, it connects to DSM Connector A or B to provide screw terminals for wiring to a third party Analog servo amplifier and I/O devices. See Figures 3-19 and 3-20.

2. In either Analog or Digital mode, it connects to DSM Connector C or D to provide screw terminals for wiring to external devices such as Strobe sensors, Home switches, and Overtravel Limit switches. See Figures 3-21, 3-22, and 3-23.

- **SL-Series Servo to APM/DSM Terminal Board, Catalog IC800SLT001** – Used to connect DSM connector A or B to a GE Fanuc SL-Series analog servo amplifier, as well as provide screw terminals for wiring to I/O devices. It contains two connectors. One connects to the DSM module, and the other to the SL-Series Servo amplifier. For additional information, please see the *SL-Series Servo User's Manual*, GFK-1581.

DSM Terminal Board Quick Selection Table			
DSM Application	**DSM Connector**	**DSM Mode**	**Terminal Board Required**
Connect to α-Series or β-Series digital servo and I/O.	A or B	Digital	IC693ACC335
Connect directly to α-Series or β-Series digital servo. No I/O connections needed.	A or B	Digital	None
Connect to third party analog servo and I/O.	A or B	Analog	IC693ACC336
Connect to SL-Series analog servo and I/O.	A or B	Analog	IC800SLT001
Connect to Auxiliary Axis I/O on DSM connector C or D	C or D	Analog or Digital	IC693ACC336

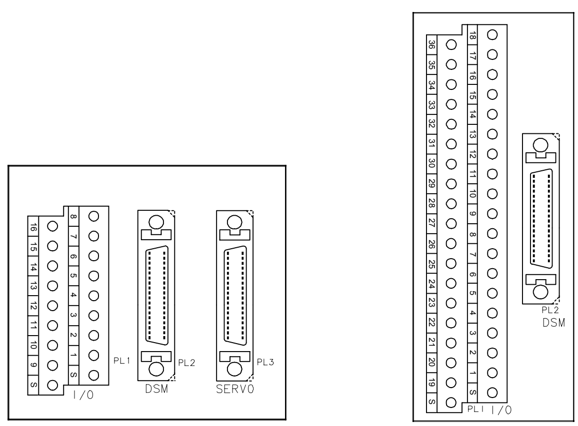


Figure 3-3. Axis and Auxiliary Terminal Board Assemblies

Note

Each Terminal Board is shipped with DIN Rail mounting feet. Instructions for converting a terminal board to panel-mount are included in this Chapter.

Digital Servo Axis Terminal Board - IC693ACC335

Description

The IC693ACC335 Digital Servo Axis Terminal Board is used to connect the DSM302 to GE Fanuc Digital Servo Amplifiers. The board contains two 36 pin connectors, labeled **DSM** and **SERVO**. A cable IC693CBL324 (1 meter) or IC693CBL325 (3 meters) connects from **DSM** connector (PL2) to the DSM302 faceplate. A Servo Command Cable IC800CBL001 (1 meter) or IC800CBL002 (3 meters) connects from the **SERVO** connector (PL3) to a GE Fanuc α series or β series Digital Servo Amplifier unit.

Eighteen screw terminals are provided on the Digital Servo Axis Terminal Board for connections to user devices. These terminals have the following assignments:

Table 3-4. Digital Axis Terminal Board Pin Assignments

Axis Terminal Board I/O Screw Terminal	DSM302 Faceplate Pin	Circuit Identifier	Circuit Type	Servo Axis 1, 2 Circuit Function	Signal Name (Axis 1 listed)*
1	1	IN1	Single ended	Strobe Input 1 (+)	IN1P_A
9	19		/ differential	Strobe Input 1 (-)	IN1M_A
2	2	IN2	5v inputs	Strobe Input 2 (+)	IN2P_A
10	20			Strobe Input 2 (-)	IN2M_A
3	4	P5V	5v Power	5v Power	P5V_A
11	22	0V	0v	0v	0V_A
6	16	IN9	24v optically	Overtravel (+)	IN9_A
14	34	IN10	isolated inputs	Overtravel (-)	IN10_A
7	17	IN11		Home Switch	IN11_A
15	35	INCOM	24v Input Common	24v Input Common	INCOM_A
8	18	OUT1	24 v, 125 ma	PLC 24v Output (+)	OUT1P_A
16	36		DC SSR output	PLC 24v Output (-)	OUT1M_A
5	14	OUT3	Differential	PLC 5v Output (+)	OUT3P_A
13	32		5v output	PLC 5v Output (-)	OUT3M_A
4	6	AOUT	+/- 10v Analog Out	PLC Analog Out	AOUT_A
12	24	ACOM	Analog Out Com	Analog Out Com	ACOM_A
S (2 pins)		SHIELD	Cable Shield	Cable Shield	SHIELD_A

* **For signal names pertaining to servo axis 2, change all "_A" to "_B".**

The maximum voltage that should be applied to I/O terminals 6-8 and 14-16 is 30 VDC. The maximum voltage for any other input terminal is 5 VDC.

Six 130V MOVs are installed between selected I/O points and the shield (frame ground) for noise suppression. The I/O terminal points so connected are 6, 7, 8, 14, 15, and 16.

The I/O terminals support a wire gauge of 14-28 AWG. Maximum screw torque which may be applied is 5 inch-pounds.

Note

Two of the screw terminals are labeled **S** for **Shield**. A short earth ground wire should be connected from one of the **S** terminals directly to a panel earth ground. The cable shields for any shielded cables from user devices should connect to either of the **S** terminals.

Mounting Dimensions

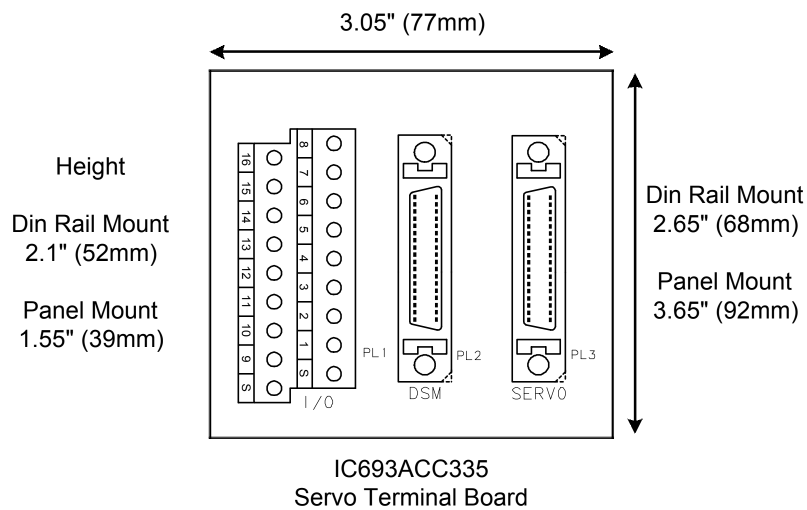


Figure 3-4. Digital Axis Terminal Board with Mounting Dimensions

Converting From DIN-Rail Mounting to Panel Mounting

The following parts are used in either the DIN-rail or Panel mount assembly options. The axis terminal board is shipped configured for DIN-rail mounting. The instructions in this section guide you in converting the board to its panel mounting optional configuration.

The following table and drawings describe the various plastic parts which make up the axis terminal board assembly and shows a side view of the board configured for DIN-rail mounting

Table 3-5. Axis Terminal Board Assembly Components

Plastic Component Part Number	Description	Quantity	Mounting Styles Used With
UMK-BE 45	Base Element	1	DIN, Panel
UMK-SE 11.25-1	Side Element	2	DIN, Panel
UMK-FE	Foot Element	2	DIN
UMK-BF*	Mounting Ear	2	Panel

* Parts shipped with axis terminal board for optional panel mounting. .

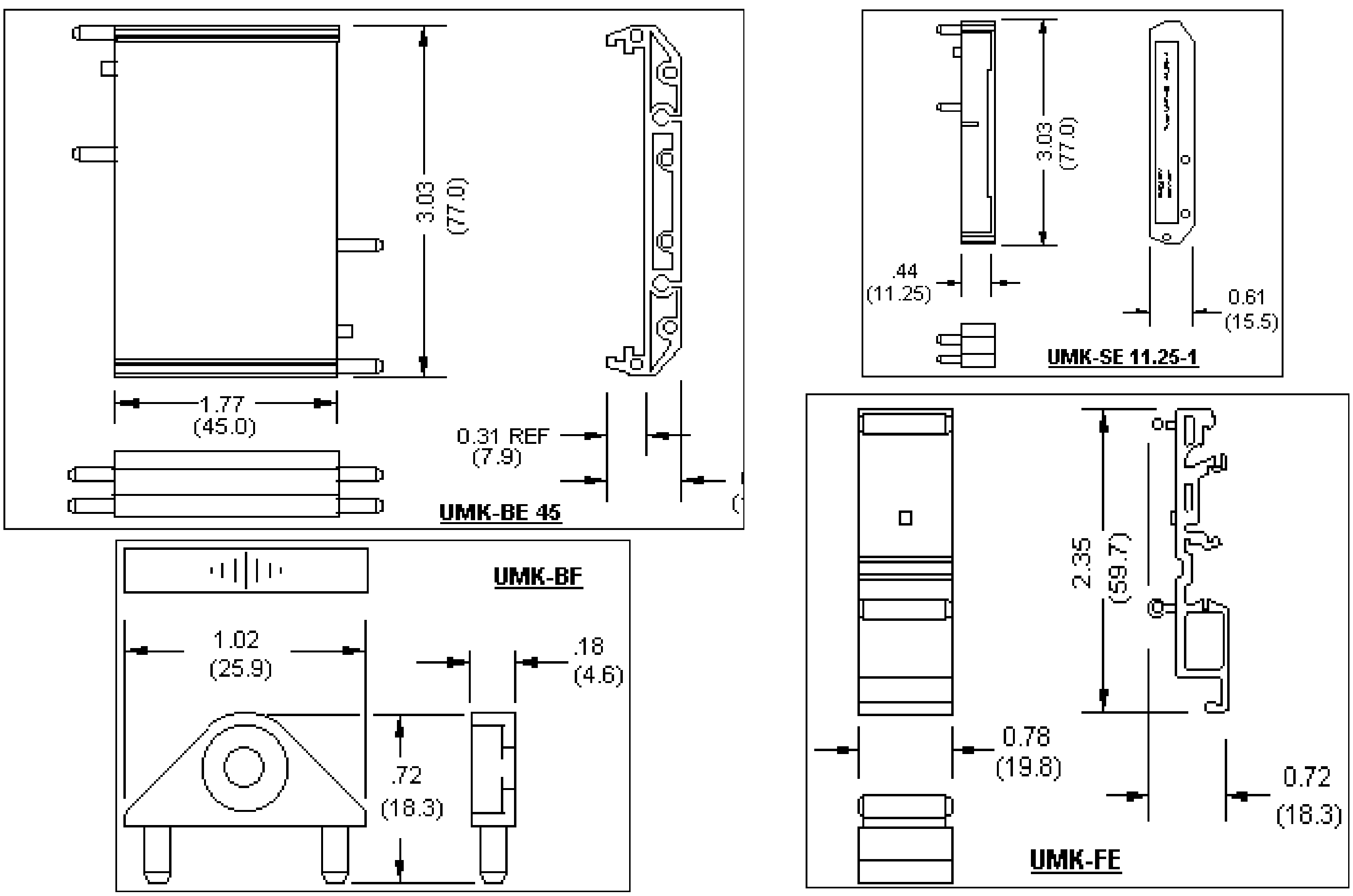


Figure 3-5. Digital Servo Axis Terminal Board Assembly Drawings

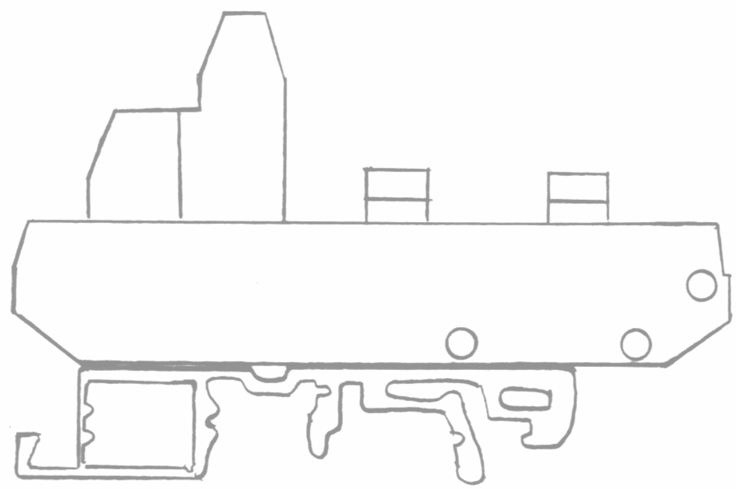


Figure 3-6. Digital Servo Axis Terminal Board Assembly Side View

The following procedure should be used to convert the Digital Servo axis terminal board to its panel mounting form. Remember to save all removed parts for possible later conversion back to DIN-rail mounting.

1. Carefully remove one UMK-SE 11.25-1 side element from the UMK-BE 45 base element. If a screwdriver or other device is used, exercise extreme caution to avoid damaging either the plastic parts or the circuit board.
2. Slide the UMK-FE foot element off the base element. Save this part for possible future use in converting the terminal board back to its DIN-rail mounting configuration.
3. Snap the side element, removed in step 1 above, back into the base element.
4. Insert one UMK-BF mounting ear into the appropriate two holes in the side element. Note that the mounting ear has a recessed hole for later inserting a (user supplied) mounting screw. The recessed hole should face <u>upwards</u> to accommodate the mounting screw.
5. Repeat steps 1-4 above for the other side of the terminal board.

Auxiliary Terminal Board - IC693ACC336

Description and Mounting Dimensions

The IC693ACC336 Auxiliary Terminal Board is used to connect the DSM302 to Analog Servo Axes and auxiliary devices such as Incremental Quadrature Encoders, Strobe detectors and external switches. The board contains one 36 pin connector, labeled **DSM**. A cable IC693CBL324 (1 meter) or IC693CBL325 (3 meters) connects from the **DSM** connector (PL2) to the DSM302 faceplate.

Thirty-eight screw terminals are provided on the Auxiliary Terminal Board for connections to user devices. These screw terminals have the same pin labels as the 36 pin DSM302 faceplate connector. Refer to Chapter 3 for detailed connection information.

The maximum voltage that should be applied to I/O terminals 16-18 and 34-36 is 30 VDC. The maximum voltage for any other input terminal is 5 VDC.

Six 130V MOVs are installed between selected I/O points and the shield (frame ground) for noise suppression. The I/O terminal points so connected are 16, 17, 18, 34, 35, and 36.

The I/O terminals support a wire gauge of 14-28 AWG. Maximum screw torque which may be applied is 5 inch-pounds.

Note

Two of the screw terminals are labeled **S** for **Shield**. A short earth ground wire should be connected from one of the **S** terminals directly to a panel earth ground. The cable shields for any shielded cables from user devices should connect to either of the **S** terminals.

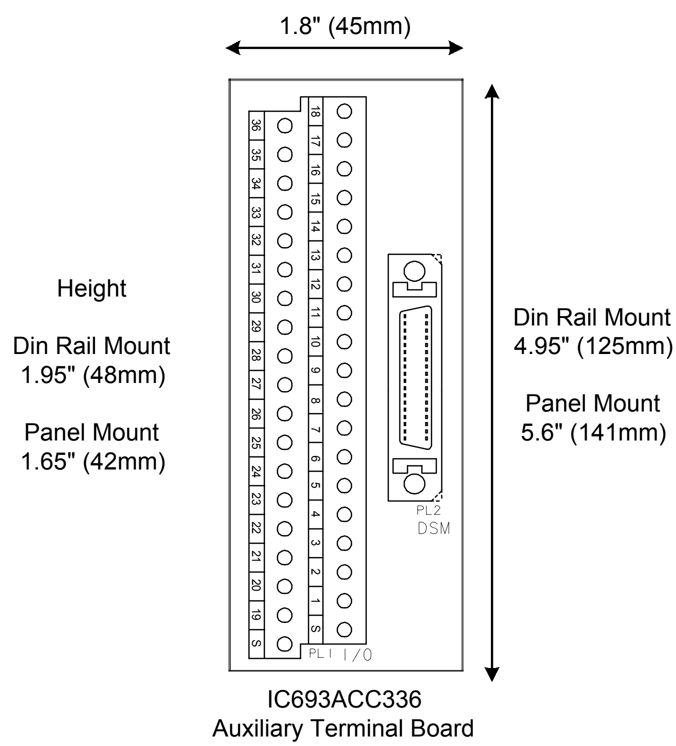


Figure 3-7. Auxiliary Terminal Board with Mounting Dimensions

Converting From DIN-Rail Mounting to Panel Mounting

The following parts are used in either the DIN-rail or Panel mount assembly options. The auxiliary terminal board is shipped configured for DIN-rail mounting. The instructions in this section guide you in converting the board to its panel mounting optional configuration.

The following table and drawings describe the various plastic parts which make up the auxiliary terminal board assembly and shows a side view of the board configured for DIN-rail mounting.

Table 3-6. Auxiliary Terminal Board Components

Phoenix Contact Part Number	Description	Quantity
UM45 Profil 105.25	PCB Carrier	1
UM 45-SEFE with 2 screws	Side element with Foot	2
UMK 45-SES with 2 screws*	Side Element	2
UMK-BF*	Mounting Ear	2

* Parts shipped with auxiliary terminal board for optional panel mounting.

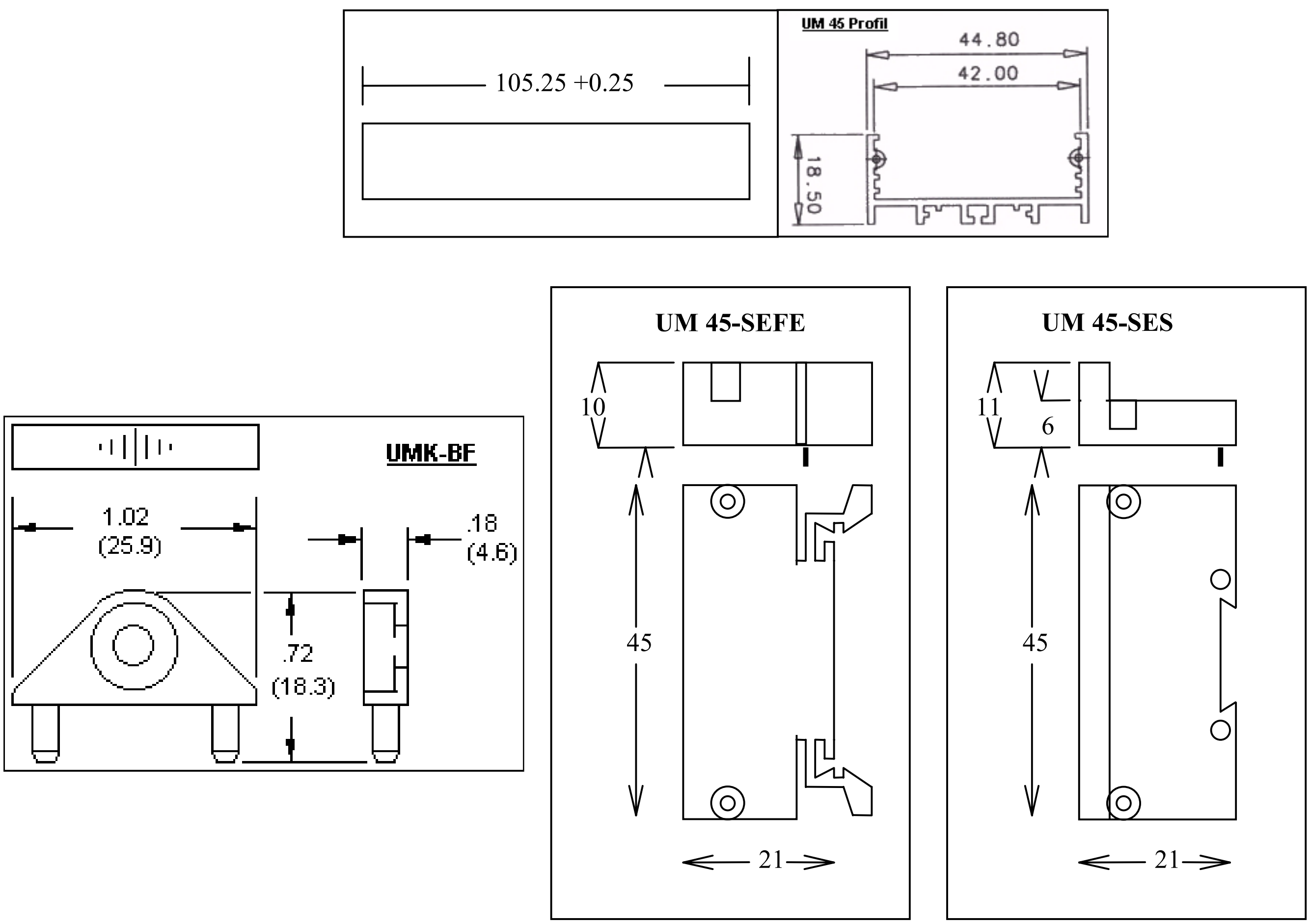


Figure 3-8. Auxiliary Terminal Board Assembly Drawings

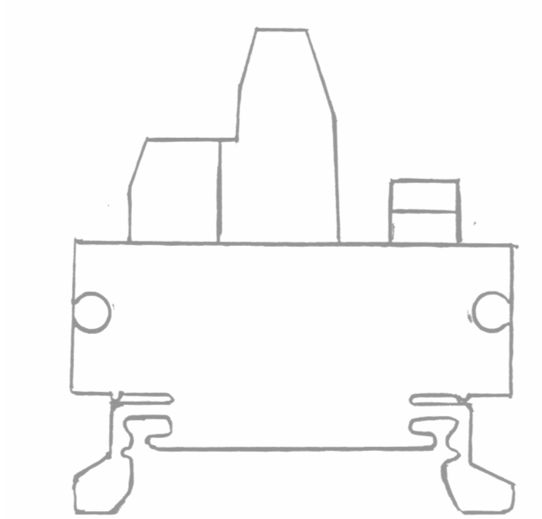


Figure 3-9. Auxiliary Terminal Board Assembly Side View

The following procedure should be used to convert the auxiliary terminal board to it panel mounting form. Remember to save all removed parts for possible later conversion back to DIN-rail mounting.

1. Using a small bladed Phillips screwdriver, carefully remove the two screws holding one UM-45 SEFE side element with foot to the UM 45 Profil PCB carrier. Save this part for possible future use in converting the terminal board back to its DIN-rail mounting configuration.
2. Attach one UMK 45-SES side element to the PCB carrier in place of the side removed in step 1 above, again using the two screws. Be careful to not overtighten the screws.
3. Insert one UMK-BF mounting ear into the appropriate two holes in the side element. Note that the mounting ear has a recessed hole for later inserting a (user supplied) mounting screw. The recessed hole should face <u>upwards</u> to accommodate the mounting screw.
4. Repeat steps 1-3 above for the other side of the terminal board.

Cables

Five cables are available for use with the DSM302:

Table 3-7. Cables for the DSM302

Cable	Description	Length	Application
IC693CBL316	Station Manager Cable	1 meter	DSM302 Comm to Motion Programmer PC
IC693CBL324	Terminal Board Connection Cable	1 meter	DSM302 to Servo Axis Terminal Board or Aux Terminal Board
IC693CBL325	Terminal Board Connection Cable	3 meters	DSM302 to Servo Axis Terminal Board or Aux Terminal Board
IC800CBL001	Digital Servo Command Cable	1 meter	Digital Servo Axis Terminal Board or DSM to Digital Servo Amp
IC800CBL002	Digital Servo Command Cable	3 meters	Digital Servo Axis Terminal Board or DSM to Digital Servo Amp

Custom Terminal Board and Servo cables are available in longer lengths by contacting your GE Fanuc distributor**.** The maximum recommended cable length for the DSM connector to the α and β Series servo amplifier is 50 meters.

The cables use special shielding and construction to ensure reliable servo operation. GE Fanuc recommends that users do not attempt any field modifications of the cables or connectors.

Note

If a Digital Servo Axis does not use any of the devices which normally connect to the IC693ACC335 Digital Servo Terminal Board screw terminals, the Terminal Board and Terminal Board Cable IC693CBL324/325 are not needed. Instead, the Digital Servo Command Cable IC800CBL001/002 can be connected directly from the Digital Servo Amplifier to the DSM302 faceplate A or B connector. When this is done, the ***OT Limit Sw*** configuration parameter must be set to DISABLE in the configuration software or the DSM will not operate.

The figure below illustrates the Digital Servo Axis terminal board and cables associated with the DSM302.

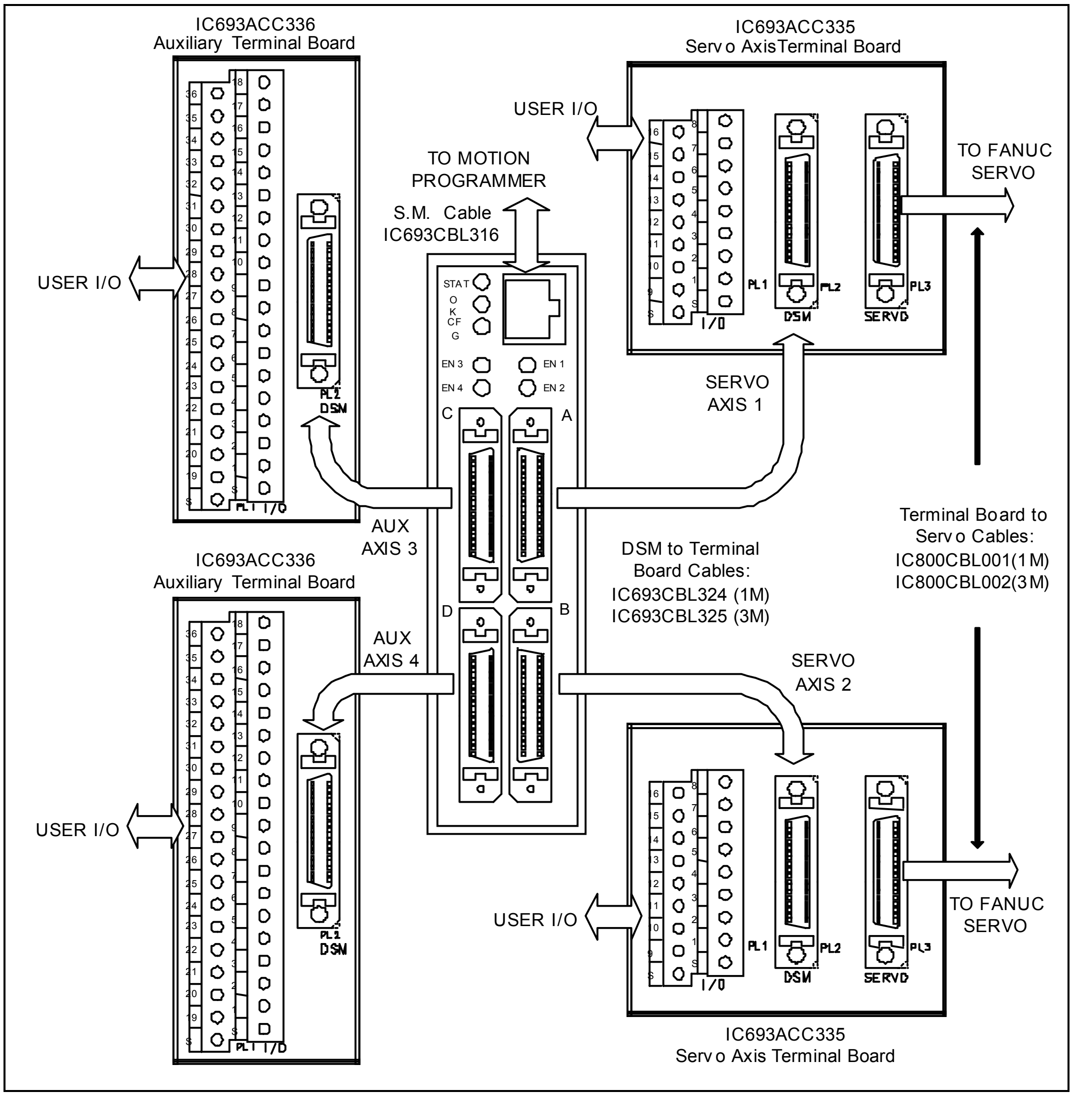


Figure 3-10. DSM302 Digital Servo Terminal Boards and Connectors

The figure below illustrates the Analog Servo terminal boards and cables associated with the DSM302.

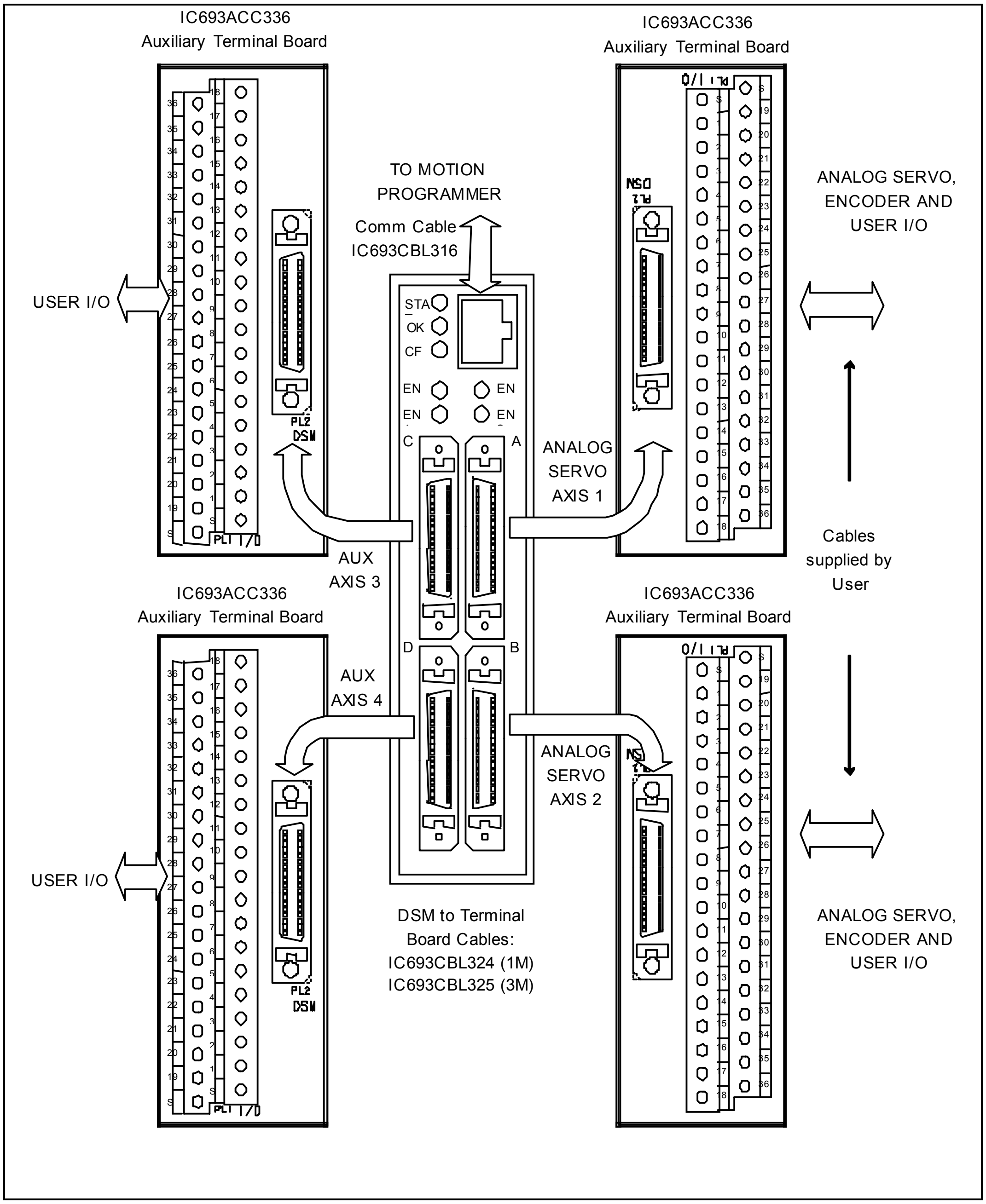


Figure 3-11. DSM302 Terminal Boards and Connectors for Analog Servos (see GFK-1581 for SL Servos)

I/O Cable Grounding

Properly routing signal cables, amplifier power cables and motor power cables along with installation of proper Class 3 grounding will insure reliable operation. Typically Class 3 grounding specifies a ground conductor of a minimum wire diameter larger than the power input wire diameter, connected via a maximum 100 ohm resistance to an earth ground. Consult local electrical codes and install in conformance to local regulations.

The specifications for completing the α and β Series Digital Servo amplifier installation and wiring, including amplifier grounding are completely described in the manual *GFH-001 Servo Product Specification Guide.*

When routing signal lines, amplifier input power line and motor power line, the signal lines must be separated from the power lines. The following table indicates how to separate the cables.

Table 3-8. Separation of signal lines

Group	Signal	Action
A	Amplifier input power Motor Power Master Control Contactor (MCC) drive coil. The MCC switches amplifier input power.	Separate a minimum 10cm from group "B" signals by bundling separately or use electromagnetic shielding (grounded steel plate). Use noise protector for MCC.
B	DSM to Axis Terminal cable Axis terminal cable to amplifier DSM to Aux Terminal cable Encoder feedback cable	Separate a minimum 10cm from group "A" signals by bundling separately or use electromagnetic shielding (grounded steel plate). Use all required individual cable shield grounds and grounding bar connections.

DSM to α or β Series Digital Servo Amplifier – Signal Cable Grounding

The signal cables used with the DSM302 contain shields which must be properly grounded to ensure reliable operation. The illustration below shows cable grounding recommendations for typical installations. The following points should be considered:

1. The DSM302 faceplate ground wire must be connected to a reliable panel ground.
2. The Digital Servo Axis Terminal Block and Auxiliary Terminal Block each provide two screw terminals labeled **S**. A short ground wire must be connected from one of the **S** terminals to a reliable panel ground.
3. The α and β Series Digital Servo amplifier encoder feedback cable always requires an A99L-0035-0001 Cable Shield Grounding Clamp and one of the 11 available slots on a 44B295864-001 Grounding Bar at the amplifier end of the cable. This clamp arrangement serves as a

mechanical strain relief and as cable shield ground. The outer insulation of the Digital servo amplifier cable must be removed to expose the cable shield in the contact area of the clamp.

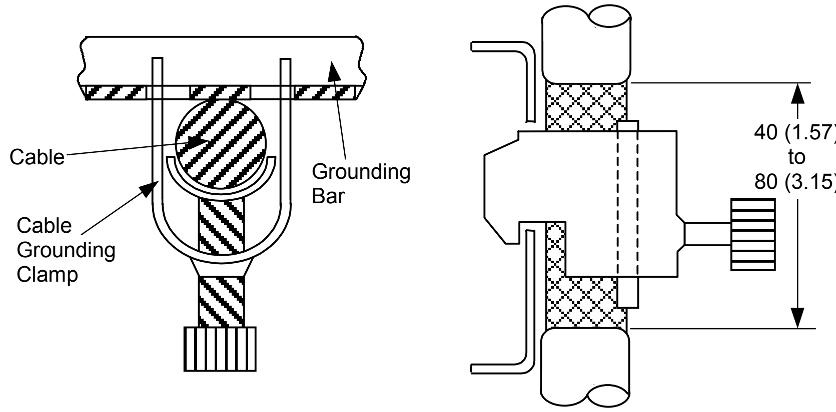


Figure 3-12. Detail of Cable Grounding Clamp A99L-0035-0001

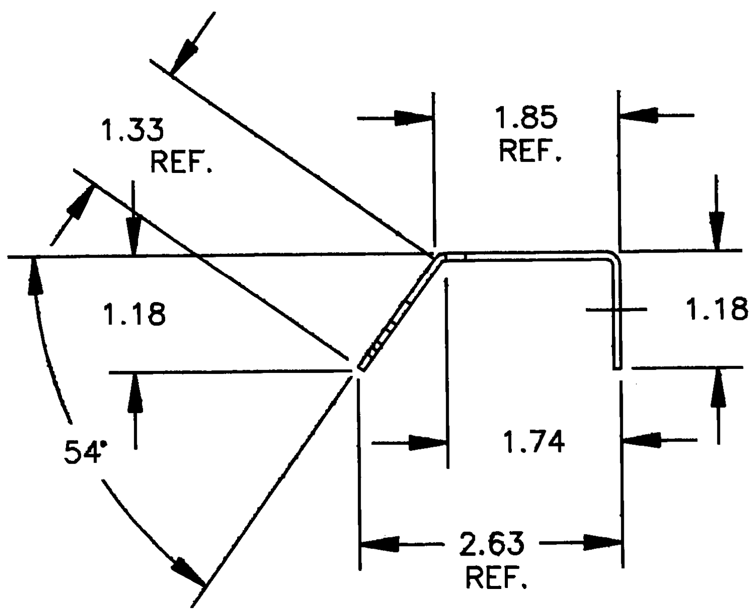


Figure 3-13. . 44B295864-001 Grounding Bar, Side View Dimensions

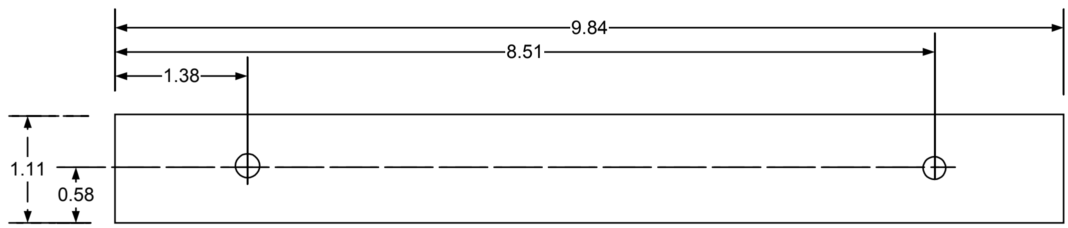


Figure 3-14. 44B295864-001 Grounding Bar Dimensions, Rear View Showing Mounting Holes

4. For installations which must meet IEC electrical noise immunity standards, a Cable Shield Grounding Clamp A99L-0035-0001 and one of the 11 available slots on the Grounding Bar 44B295864-001 must also be used at the Digital Servo Axis Terminal Block end of the servo amplifier cable IC800CBL001/002. If the Digital servo amplifier cable is connected directly to the DSM302 faceplate (no Digital Servo Axis Terminal Block used) the Grounding Clamp and Bar are not required at the faceplate end of the cable.

For additional information, refer to *Installation Requirements for Conformance to Standards*, GFK-1179.

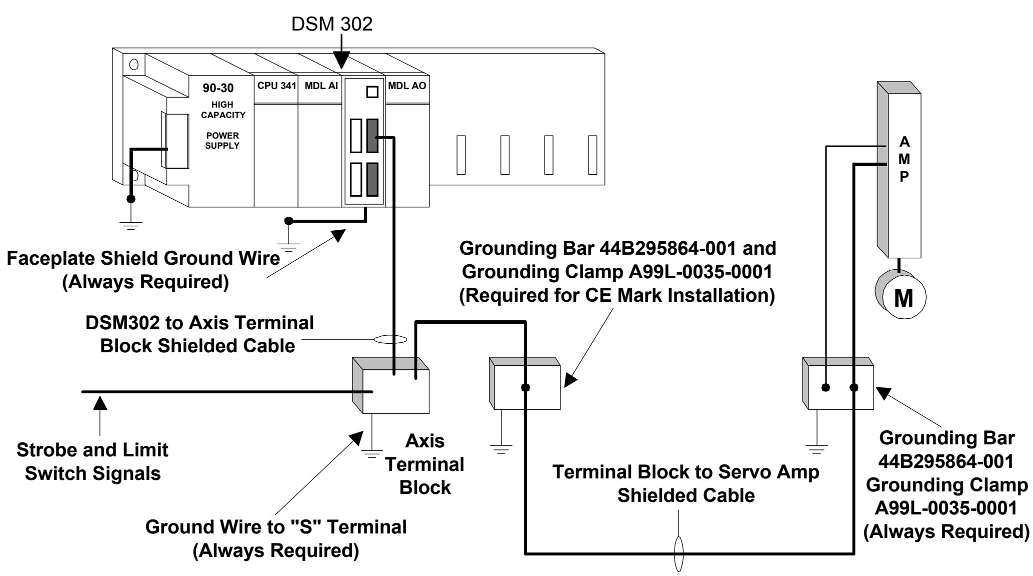


Figure 3-15. DSM302 I/O Cable Grounding

I/O Circuit Identifiers and Signal Names

I/O circuit identifiers provide a consistent method of naming the I/O circuits. For example, IN1 refers to the first of three differential / single ended 5v inputs for each axis.

Signal names are assigned to the circuit identifiers for each axis. *The signal name consists of the circuit identifier followed by a suffix A-D to identify the axis connector*. Differential circuits also have suffixes P (positive) and M (minus) to identify the (+) and (-) signal for each differential pair.

Example: OUT2 is the circuit identifier for the first differential 5v output on each connector. The signal names associated with circuit OUT2 are:

Table 3-9. Signal Names Associated with OUT2

Axis:	**Servo Axis 1**	**Servo Axis 2**	**Aux Axis 3**	**Aux Axis 4**
Connector:	A	B	C	D
(+) Output Signal:	OUT2P_A	OUT2P_B	OUT2P_C	OUT2P_D
(-) Output Signal:	OUT2M_A	OUT2M_B	OUT2M_C	OUT2M_D

I/O Circuit Function and Pin Assignments

The next three tables list the I/O circuit functional assignments as well as the connector and terminal board pin assignments for each axis connector. Although each connector has the same I/O circuits, the functional assignment of the I/O circuits is axis dependent:

Table 3-10. Connector Axis Assignment and Function

Connector	**Axis Number**	**Axis Type**	**I/O Usage**
A	1	Servo Axis	Closed Loop Servo Control and user I/O
B	2	Servo Axis	Closed Loop Servo Control and user I/O
C	3	Aux Axis	Master Position Feedback and auxiliary analog and digital I/O
D	4	Aux Axis	Auxiliary analog and digital I/O

Digital Servo Axis 1, 2 Circuit and Pin Assignments

This table identifies all circuits and pin assignments for Digital Servo Axis 1 and Digital Servo Axis 2. *The shaded areas indicate signals which are cabled to the servo amplifier and are not available for user connections.*

Table 3-11. Circuit and Pin Assignments for Digital Servo Axis 1 and Digital Servo Axis 2

Circuit Identifier	Circuit Type	Servo Axis 1, 2 Circuit Function	Axis 1 Signal Name	Axis 2 Signal Name	Faceplate Pin	Axis Term Board Terminal
IN1	Single ended / differential 5v inputs	Strobe Input 1 (+) Strobe Input 1 (-)	IN1P_A IN1M_A	IN1P_B IN1M_B	1 19	1 9
IN2		Strobe Input 2 (+) Strobe Input 2 (-)	IN2P_A IN2M_A	IN2P_B IN2M_B	2 20	2 10
IN3		Ser Encoder Data (+) Ser Encoder Data (-)	IN3P_A IN3M_A	IN3P_B IN3M_B	3 21	
P5V	5v Power	5v Power	P5V_A	P5V_B	4	3
0V	0v	0v	0V_A	0V_B	22,23	11
IN4	Single ended 5v in	Servo Ready Input	IN4_A	IN4_B	5	
IO5	Single ended 5v inputs / outputs	Servo PWM / Alarm	IO5_A	IO5_B	9	
IO6		Servo PWM / Alarm	IO6_A	IO6_B	10	
IO7		Servo PWM / Alarm	IO7_A	IO7_B	11	
IO8		Servo ENBL / Alarm	IO8_A	IO8_B	12	
0V	0v	0v	0V_A	0V_B	27-30	
IN9	24v optically isolated inputs	Overtravel (+)	IN9_A	IN9_B	16	6
IN10		Overtravel (-)	IN10_A	IN10_B	34	14
IN11		Home Switch	IN11_A	IN11_B	17	7
INCOM	24v Input Common	24v Input Common	INCOM_A	INCOM_B	35	15
OUT1	24 v, 125 ma DC SSR output	PLC 24v Output (+) PLC 24v Output (-)	OUT1P_A OUT1M_A	OUT1P_B OUT1M_B	18 36	8 16
OUT2	Differential 5v outputs	Ser Encoder Req (+) Ser Encoder Req (-)	OUT2P_A OUT2M_A	OUT2P_B OUT2M_B	13 31	
OUT3		PLC 5v Output (+) PLC 5v Output (-)	OUT3P_A OUT3M_A	OUT3P_B OUT3M_B	14 32	5 13
ENBL	24v, 30 ma SSR output	Servo MCON (+) Servo MCON 0v	ENBL1_A ENBL2_A	ENBL1_B ENBL2_B	15 33	
AIN1	Differential +/- 10v Analog Inputs	IR Phase Current (+) IR Phase Current (-)	AIN1P_A AIN1M_A	AIN1P_B AIN1M_B	7 25	
AIN2		IS Phase Current (+) IS Phase Current (-)	AIN2P_A AIN2M_A	AIN2P_B AIN2M_B	8 26	
AOUT1	+/- 10v Analog Out	PLC Analog Out	AOUT_A	AOUT_B	6	4
ACOM	Analog Out com	Analog Out Com	ACOM_A	ACOM_B	24	12
SHIELD	Cable Shield	Cable Shield	SHIELD_A	SHIELD_B		S

Analog Servo Axis 1, 2 Circuit and Pin Assignments

This table identifies all circuits and pin assignments for Analog Servo Axis 1 and Analog Servo Axis 2. *The shaded areas indicate signals which are unused and not available for user connections.*

Table 3-12. Circuit and Pin Assignments for Analog Servo Axis 1 and Analog Servo Axis 2

Circuit Identifier	Circuit Type	Servo Axis 1, 2 Circuit Function	Axis 1 Signal Name	Axis 2 Signal Name	Faceplate Pin	Aux Term Board Terminal
IN1	Single ended / differential 5v inputs	Encoder Chan A (+)	IN1P_A	IN1P_B	1	1
		Encoder Chan A (-)	IN1M_A	IN1M_B	19	19
IN2		Encoder Chan B (+)	IN2P_A	IN2P_B	2	2
		Encoder Chan B (-)	IN2M_A	IN2M_B	20	20
IN3		Encoder Marker (+)	IN3P_A	IN3P_B	3	3
		Encoder Marker (-)	IN3M_A	IN3M_B	21	21
P5V	5v Power	5v Encoder Power	P5V_A	P5V_B	4	4
0V	0v	0v	0V_A	0V_B	22,23	22,23
IN4	Single ended 5v in	Servo Ready Input	IN4_A	IN4_B	5	5
IO5	Single ended 5v inputs / outputs	Strobe 1 Input	IO5_A	IO5_B	9	9
IO6		Strobe 2 Input	IO6_A	IO6_B	10	10
IO7		Not Used	IO7_A	IO7_B	11	11
IO8		Not Used	IO8_A	IO8_B	12	12
0V	0v	0v	0V_A	0V_B	27-30	27-30
IN9	24v optically isolated inputs	Overtravel (+)	IN9_A	IN9_B	16	16
IN10		Overtravel (-)	IN10_A	IN10_B	34	34
IN11		Home Switch	IN11_A	IN11_B	17	17
INCOM	24v Input Common	24v Input Common	INCOM_A	INCOM_B	35	35
OUT1	24 v, 125 ma DC SSR output	PLC 24v Output (+)	OUT1P_A	OUT1P_B	18	18
		PLC 24v Output (-)	OUT1M_A	OUT1M_B	36	36
OUT2	Differential 5v outputs	Not Used	OUT2P_A	OUT2P_B	13	13
		Not Used	OUT2M_A	OUT2M_B	31	31
OUT3		PLC 5v Output (+)	OUT3P_A	OUT3P_B	14	14
		PLC 5v Output (-)	OUT3M_A	OUT3M_B	32	32
ENBL	24v, 30 ma SSR output	Servo Enable (+)	ENBL1_A	ENBL1_B	15	15
		Servo Enable (-)	ENBL2_A	ENBL2_B	33	33
AIN1	Differential +/- 10v Analog Inputs	PLC Analog In (+)	AIN1P_A	AIN1P_B	7	7
		PLC Analog In (-)	AIN1M_A	AIN1M_B	25	25
AIN2		PLC Analog In (+)	AIN2P_A	AIN2P_B	8	8
		PLC Analog In (-)	AIN2M_A	AIN2M_B	26	26
AOUT1	+/- 10v Analog Out	Servo Vel Cmd (+)	AOUT_A	AOUT_B	6	6
ACOM	Analog Out com	Servo Vel Cmd Com	ACOM_A	ACOM_B	24	24
SHIELD	Cable Shield	Cable Shield	SHIELD_A	SHIELD_B		S

Aux Axis 3 (Follower Master Axis) Circuit and Pin Assignments

This table identifies all circuits and pin assignments for Aux Axis 3 (Connector C). *The shaded areas indicate signals, that are unused and not available for user connections*.

Table 3-13. Circuit and Pin Assignments for Aux Axis 3 (Connector C)

Circuit Identifier	Circuit Type	Aux Axis 3 Circuit Function	Axis 3 Signal Name	Faceplate Pin	Aux Term Board Terminal
IN1	Single ended / differential 5v inputs	Encoder Chan A (+) Encoder Chan A (-)	IN1P_C IN1M_C	1 19	1 19
IN2		Encoder Chan B (+) Encoder Chan B (-)	IN2P_C IN2M_C	2 20	2 20
IN3		Encoder Marker (+) Encoder Marker (-)	IN3P_C IN3M_C	3 21	3 21
P5V	5v from PLC	5v Encoder Power	P5V_C	4	4
0V	0v	0v	0V_C	22,23	22,23
IN4	Single ended 5v in	PLC 5v Input	IN4_C	5	5
IO5	Single ended 5v inputs / outputs	Strobe 1 Input	IO5_C	9	9
IO6		Strobe 2 Input	IO6_C	10	10
IO7		Not Used	IO7_C	11	11
IO8		Not Used	IO8_C	12	12
0V	0v	0v	0V_C	27-30	27-30
IN9	24v optically isolated inputs	PLC 24v Input	IN9_C	16	16
IN10		PLC 24v Input	IN10_C	34	34
IN11		Home Switch	IN11_C	17	17
INCOM	24v Input Common	24v Input Common	INCOM_C	35	35
OUT1	24 v, 125 ma DC SSR output	PLC 24v Output (+) PLC 24v Output (-)	OUT1P_C OUT1M_C	18 36	18 36
OUT2	Differential 5v outputs	Not Used Not Used	OUT2P_C OUT2M_C	13 31	13 31
OUT3		PLC 5v Output (+) PLC 5v Output (-)	OUT3P_C OUT3M_C	14 32	14 32
ENBL	24v, 30 ma SSR output	ON when Force Analog Output %AQ Cmd is active	ENBL1_C ENBL2_C	15 33	15 33
AIN1	Differential +/- 10v Analog Inputs	PLC Analog In (+) PLC Analog In (-)	AIN1P_C AIN1M_C	7 25	7 25
AIN2		PLC Analog In (+) PLC Analog In (-)	AIN2P_C AIN2M_C	8 26	8 26
AOUT1	+/- 10v Analog Out	PLC Analog Out	AOUT_C	6	6
ACOM	Analog Out com	Analog Out Com	ACOM_C	24	24
SHIELD	Cable Shield	Cable Shield	SHIELD_C		S

Aux Axis 4 Circuit and Pin Assignments

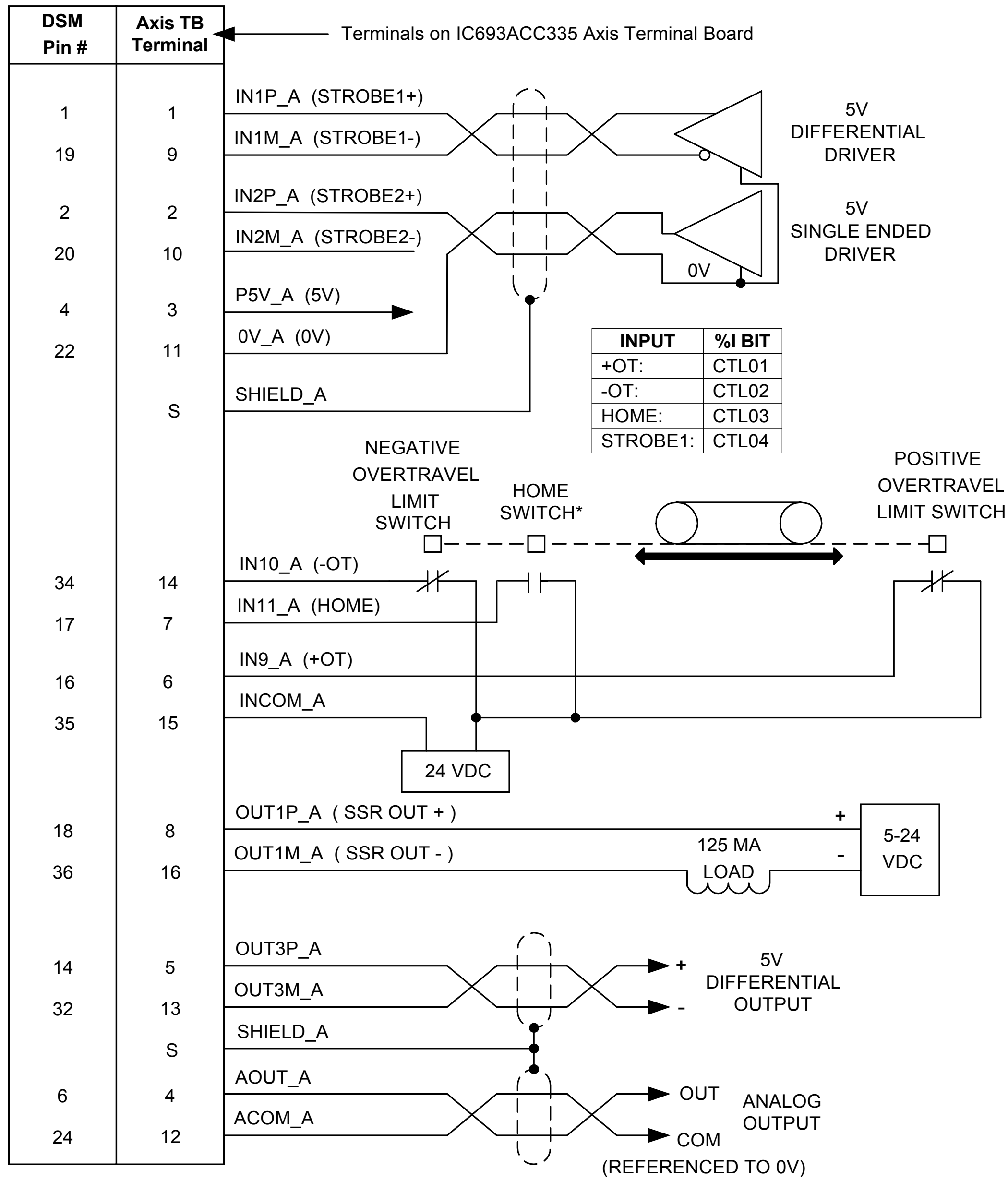
This table identifies all circuits and pin assignments for Aux Axis 4 (Connector D). *The shaded areas indicate signals which are unused and not available for user connections*.

Table 3-14. Circuit and Pin Assignments for Aux Axis 4 (Connector D)

Circuit Identifier	Circuit Type	Aux Axis 4 Circuit Function	Axis 4 Signal Name	Faceplate Pin	Aux Term Board Terminal
IN1	Single ended / differential 5v inputs	Not Used Not Used	IN1P_D IN1M_D	1 19	1 19
IN2		Not Used Not Used	IN2P_D IN2M_D	2 20	2 20
IN3		Not Used Not Used	IN3P_D IN3M_D	3 21	3 21
P5V	5v from PLC	5v Power	P5V_D	4	4
0V	0v	0v	0V_D	22,23	22,23
IN4	Single ended 5v in	Not Used	IN4_D	5	5
IO5	Single ended 5v inputs / outputs	PLC 5v Input	IO5_D	9	9
IO6		Not Used	IO6_D	10	10
IO7		Not Used	IO7_D	11	11
IO8		Not Used	IO8_D	12	12
0V	0v	0v	0V_D	27-30	27-30
IN9	24v optically isolated inputs	PLC 24v Input	IN9_D	16	16
IN10		PLC 24v Input	IN10_D	34	34
IN11		PLC 24v Input	IN11_D	17	17
INCOM	24v Input Common	24v Input Common	INCOM_D	35	35
OUT1	24 v, 125 ma DC SSR output	PLC 24v Output (+) PLC 24v Output (-)	OUT1P_D OUT1M_D	18 36	18 36
OUT2	Differential 5v outputs	Not Used Not Used	OUT2P_D OUT2M_D	13 31	13 31
OUT3		PLC 5v Output (+) PLC 5v Output (-)	OUT3P_D OUT3M_D	14 32	14 32
ENBL	24v, 30 ma SSR output	ON when Force Analog Output %AQ Cmd is active	ENBL1_D ENBL2_D	15 33	15 33
AIN1	Differential +/- 10v Analog Inputs	PLC Analog In (+) PLC Analog In (-)	AIN1P_D AIN1M_D	7 25	7 25
AIN2		PLC Analog In (+) PLC Analog In (-)	AIN2P_D AIN2M_D	8 26	8 26
AOUT1	+/- 10v Analog Out	PLC Analog Out	AOUT_D	6	6
ACOM	Analog Out com	Analog Out Com	ACOM_D	24	24
SHIELD	Cable Shield	Cable Shield	SHIELD_D		S

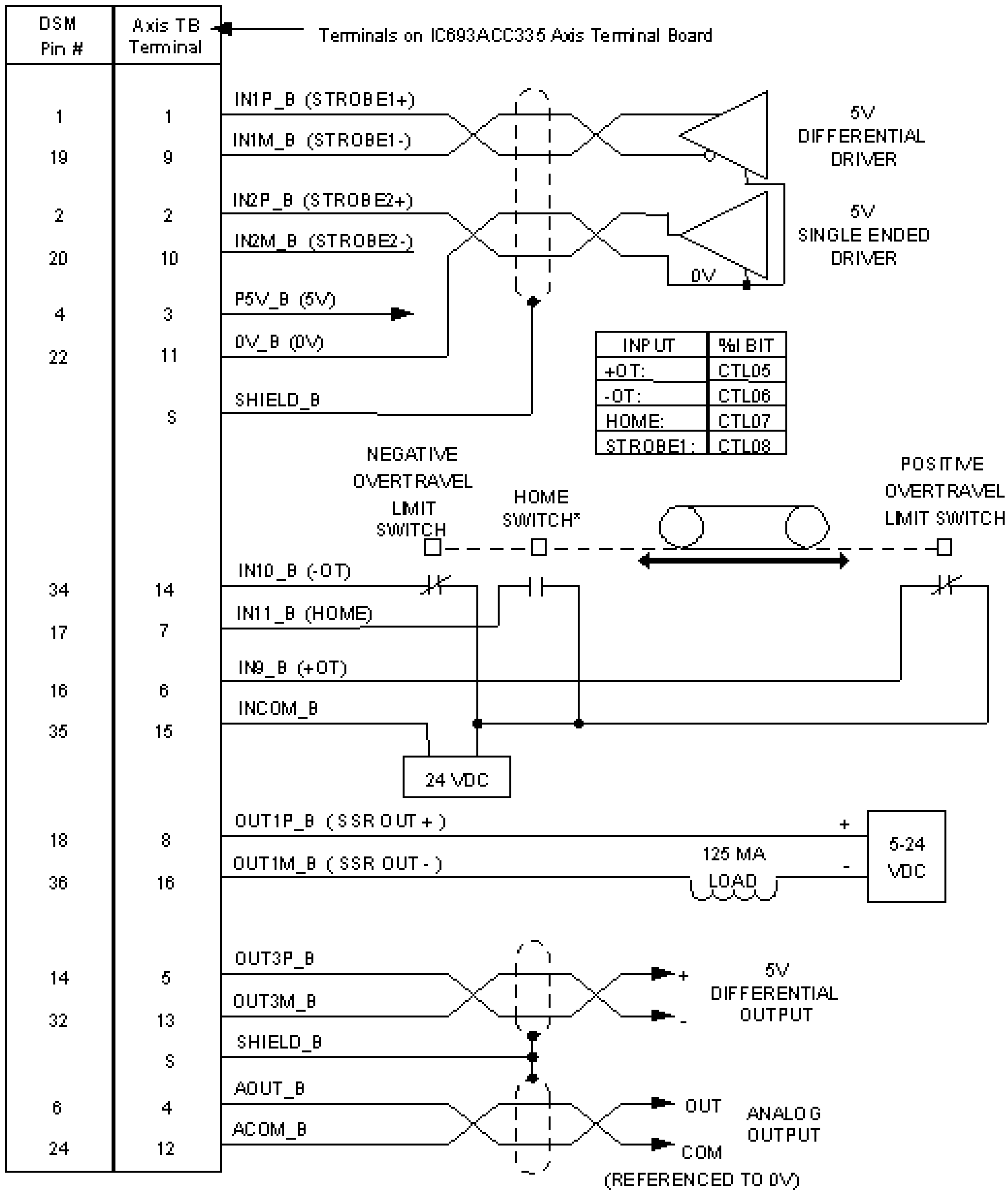
I/O Connection Diagrams

The following diagrams illustrate typical user connections to the DSM302.



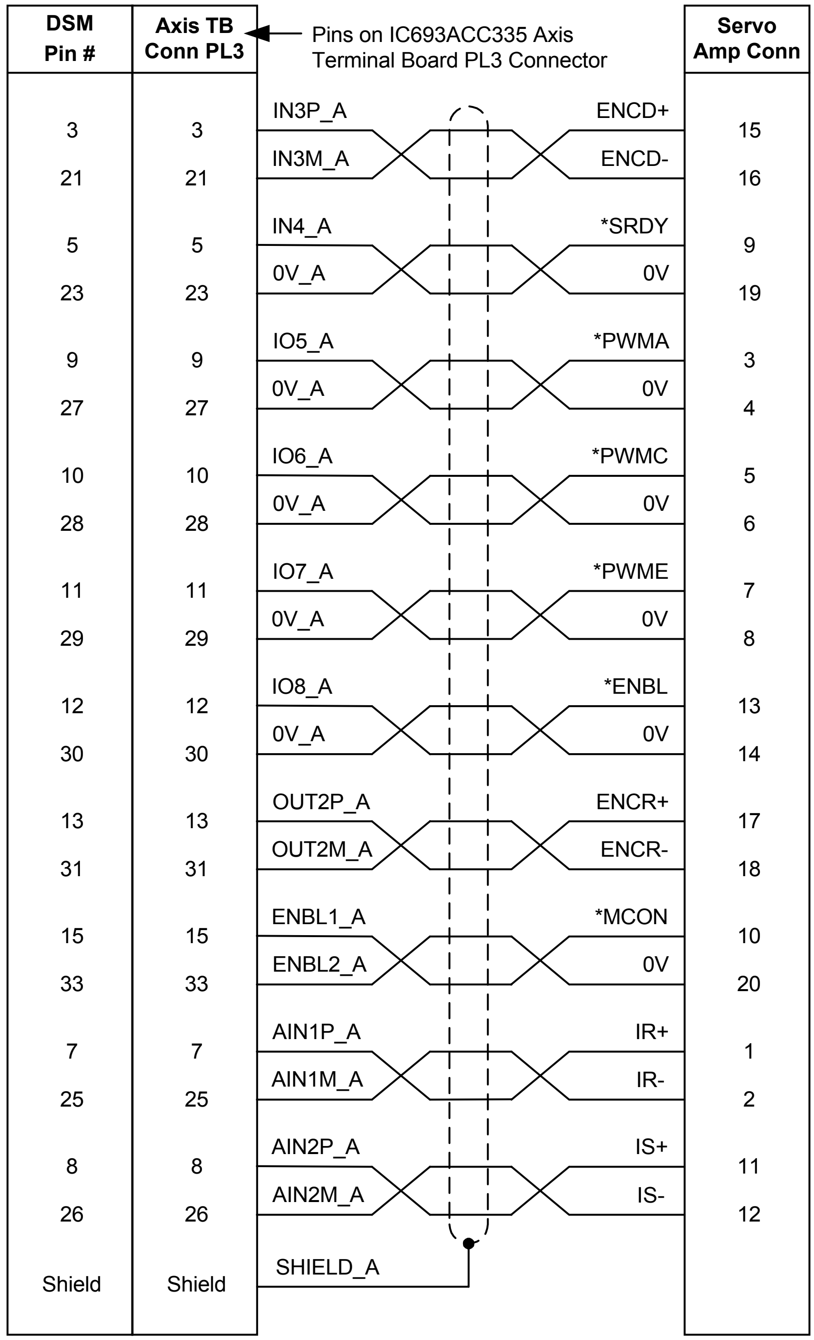
* Note: See Chapter 6 for home switch information

Figure 3-16. Digital Servo Axis-1 Connections



*Note: See Chapter 6 for home switch information

Figure 3-17. Digital Servo Axis-2 Connections



* Denotes a negated signal

Figure 3-18. α and β Series Digital Servo Command Cable (IC800CBL001/002) Connections

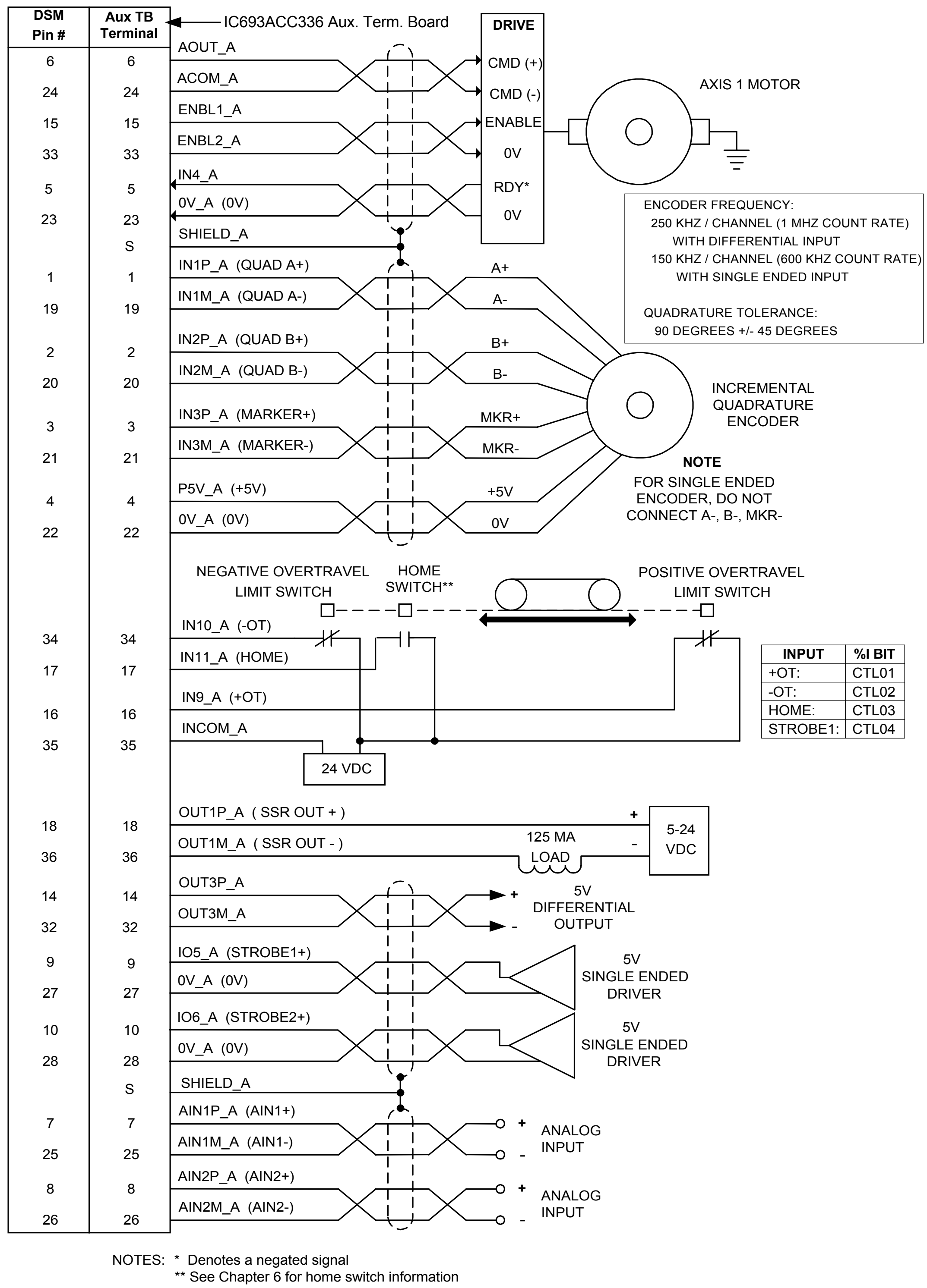


Figure 3-19. Analog Servo Axis-1 Connections

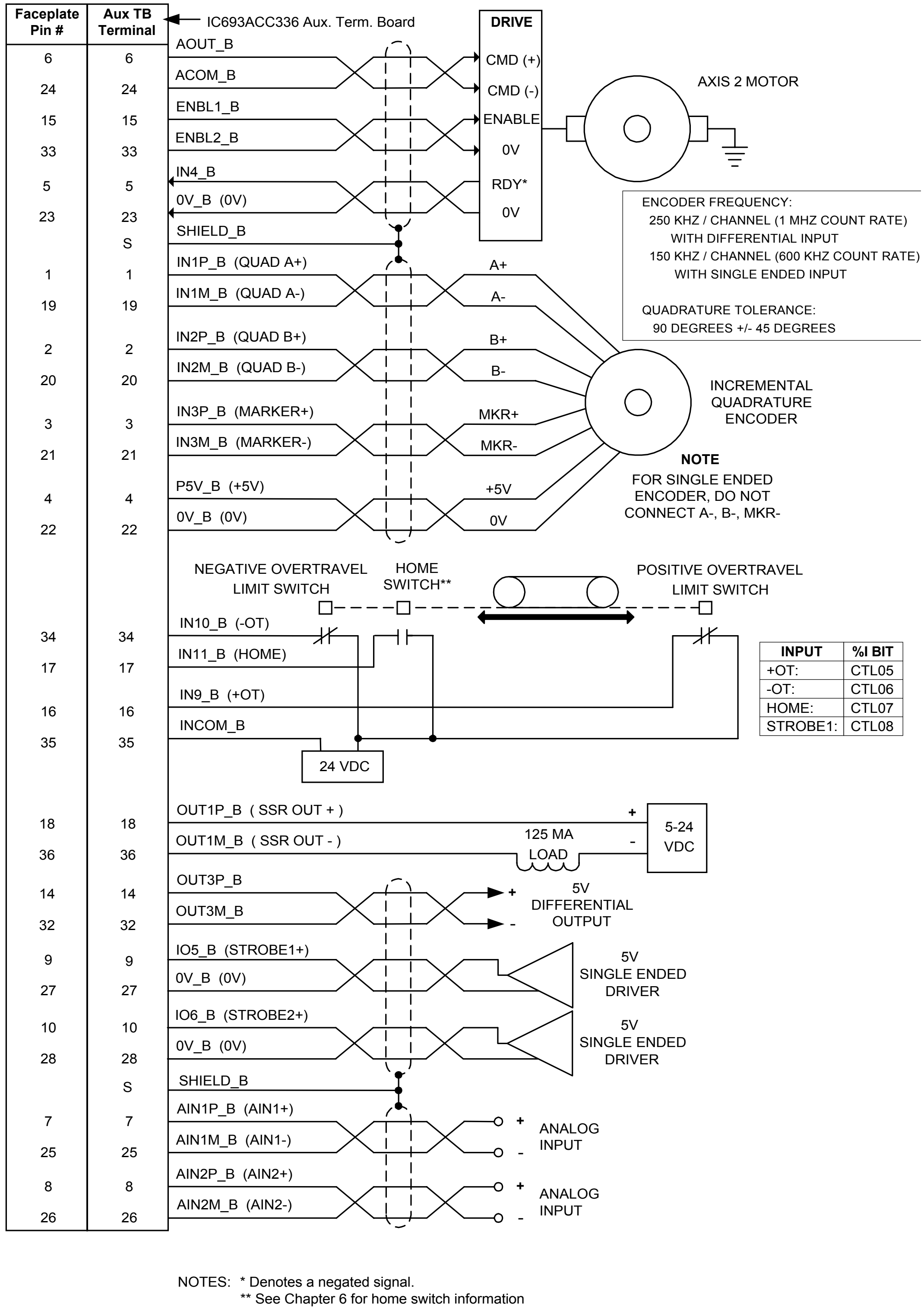


Figure 3-20. Analog Servo Axis-2 Connections

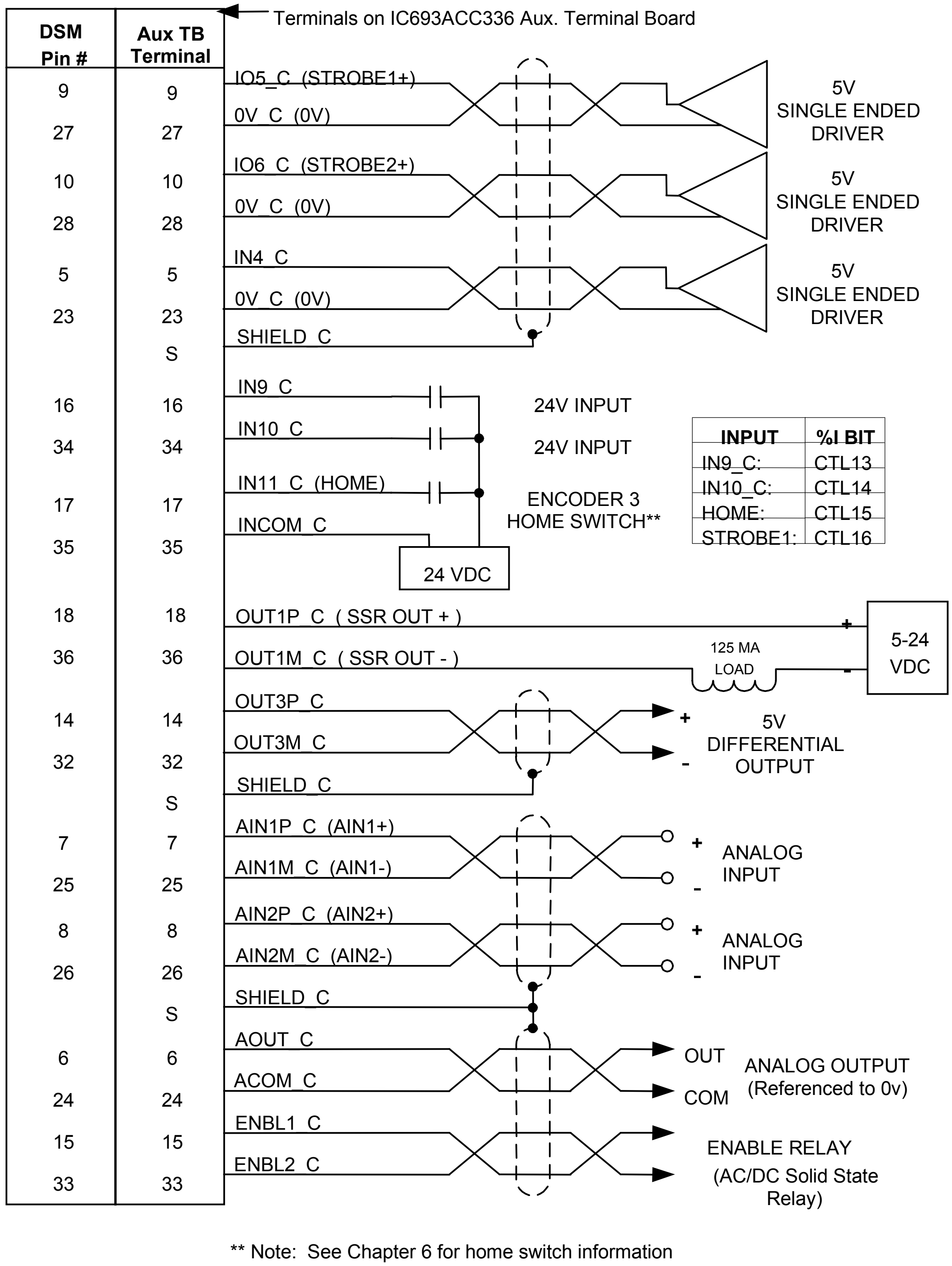


Figure 3-21. Aux Axis-3 (Follower Master Axis) Connections

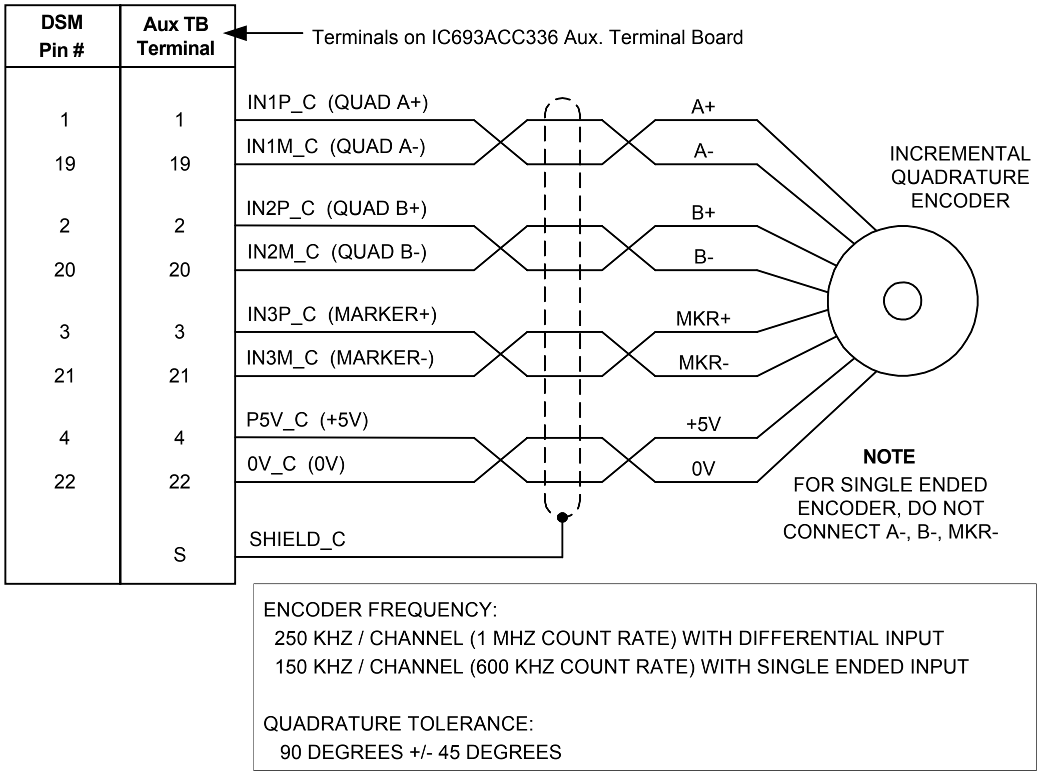


Figure 3-22. Aux Axis-3 (Follower Master Axis) Encoder Connections

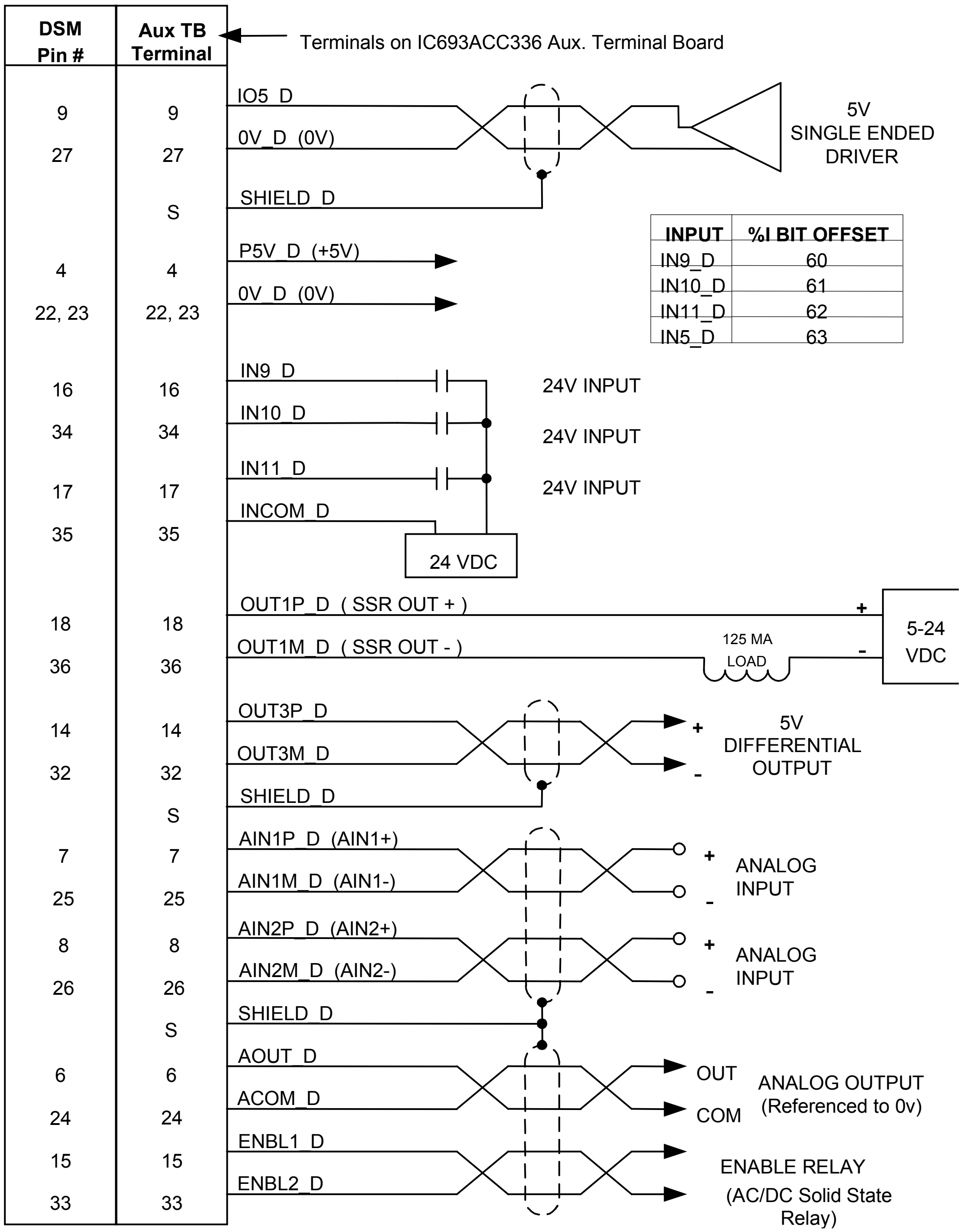


Figure 3-23. Aux Axis-4 Connections

I/O Specifications

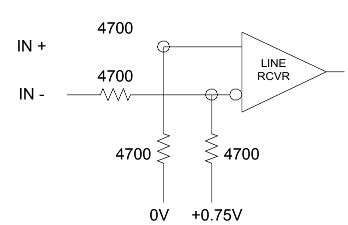
The specifications and simplified schematics for the module's I/O circuits are provided on the following pages. The I/O circuits described are as follows:

- Differential/Single Ended 5v Inputs (IN1, IN2, IN3)
- Single Ended 5v Sink Input (IN4)
- Optically Isolated 24v Source/Sink Inputs (IN9, IN10, IN11, INCOM)
- Single Ended 5v Inputs/Outputs (IO5, IO6, IO7, IO8)
- 5v Differential Outputs (OUT2, OUT3)
- 24v DC Optically Isolated Output (OUT1)
- Optically Isolated Enable Relay Output (OUT4)
- Differential +/- 10v Analog Inputs (AIN1, AIN2)
- Single Ended +/- 10v Analog Outputs (AOUT1, ACOM)
- +5v Power (P5V, 0V)

Differential / Single Ended 5v Inputs

Circuit Identifier	*Digital Servo Axis 1, 2 Circuit Function*	*Analog Servo Axis 1, 2 and Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
IN1	Strobe Input 1 (+) Strobe Input 1 (-)	Encoder Chan. A (+) Encoder Chan. A (-)	Not Used Not Used	IN1P_A IN1M_A	1 19	1 19	1 9
IN2	Strobe Input 2 (+) Strobe Input 2 (-)	Encoder Chan. B (+) Encoder Chan. B (-)	Not Used Not Used	IN2P_A IN2M_A	2 20	2 20	2 10
IN3	Ser Encoder Data (+) Ser Encoder Data (-)	Encoder Marker (+) Encoder Marker (-)	Not Used Not Used	IN3P_A IN3M_A	3 21	3 21	N/C N/C

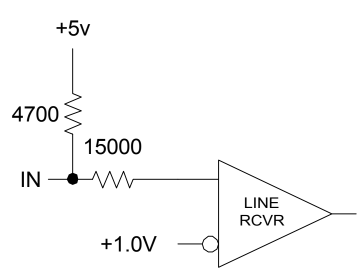
I/O Type:	Differential / Single Ended 5v Inputs
Circuit Type:	Source Input (9.4K ohm pulldown to 0v)
Input Impedance: (+) or (-) Input	9.4K ohms common mode to 0v 18.8K ohms differential
Maximum Input Voltage:	+/- 15 v common mode +/- 20 v differential
Logic 0 Threshold:	+0.8 v max single ended +0.4 v max differential
Logic 1 Threshold :	+2.0 v min single ended +1.5 v min differential
Input Filtering:	0.5 microseconds typical
Quadrature Encoder Frequency:	250 khz/channel (1 mhz count rate) max with differential inputs 150 khz/channel (600 khz count rate) max with single ended inputs Quadrature Tolerance: 90 degrees +/- 45 degrees
Strobe Response:	Minimum Pulse Width: 3 microseconds Position Capture Delay: 10 to 260 microseconds
Notes:	Use (+) Input for single ended mode and leave (-) input floating. Use faceplate 0v pins for common mode reference or single ended signal return. Inputs can be driven by 5v TTL or CMOS logic.



Single Ended 5v Sink Input

Circuit Identifier	*Servo Axis 1, 2 Circuit Function*	*Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
IN4	Servo Ready Input	Faceplate 5v Input	Not Used	IN4_A	5	5	N/C

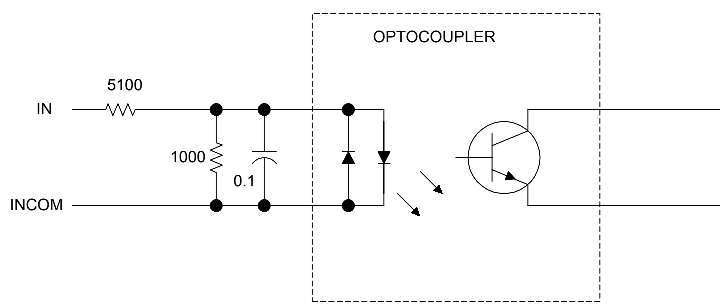
I/O Type:	Single Ended 5v Sink Input
Circuit Type:	Sink Input (4.7K ohm pull-up to internal +5v)
Input Impedance:	4.7K ohms to +5v
Maximum Input Voltage:	+/- 10.0 v
Logic 0 Threshold:	+0.8 v max
Logic 1 Threshold :	+2.0 v min
Input Filtering:	1.0 microseconds typical
Notes:	This input must be pulled to 0v to turn on.



Optically Isolated 24v Source / Sink Inputs

Circuit Identifier	*Servo Axis 1, 2 Circuit Function*	*Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
IN9	Overtravel (+)	Faceplate 24v Input	Faceplate 24v Input	IN9_A	16	16	6
IN10	Overtravel (-)	Faceplate 24v Input	Faceplate 24v Input	IN10_A	34	34	14
IN11	Home Switch	Home Switch	Faceplate 24v Input	IN11_A	17	17	7
INCOM	24v Input Common	24v Input Common	24v Input Common	INCOM_A	35	35	15

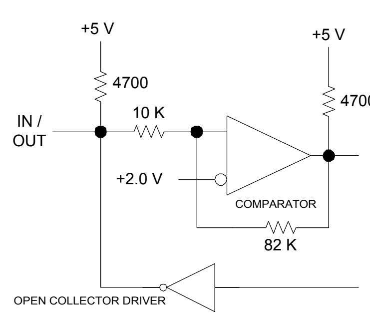
I/O Type:	Optically Isolated 24v Source / Sink Inputs
Circuit Type:	Source / Sink (5K resistance to INCOM)
Input Impedance:	5.4K ohms to INCOM (@ 24 VDC)
Maximum Input Voltage:	+/- 30.0v (referenced to INCOM)
Logic 0 Threshold:	+/- 6.0 v max (referenced to INCOM)
Logic 1 Threshold :	+/- 18.0 v min (referenced to INCOM)
Input Filtering:	5 milliseconds typical
Notes:	These inputs use bi-directional optocouplers and can be turned on with either a positive or negative input with respect to INCOM.



Single Ended 5v Inputs/Outputs

Circuit Identifier	*Digital Servo Axis 1, 2 Circuit Function*	*Analog Servo Axis 1, 2 and Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
IO5 0V	Servo PWM / Alarm 0v	Strobe 1 Input 0v	Not Used 0v	IO5_A/IN5_A 0V_A	9 27	9 27	N/C N/C
IO6 0V	Servo PWM / Alarm 0v	Strobe 2 Input 0v	Not Used 0v	IO6_A/IN6_A 0V_A	10 28	10 28	N/C N/C
IO7 0V	Servo PWM / Alarm 0v	Not Used 0v	Not Used 0v	IO7_A/IN7_A 0V_A	11 29	11 29	N/C N/C
IO8 0V	Servo ENBL / Alarm 0v	Not Used 0v	Not Used 0v	IO8_A/IN8_A 0V_A	12 30	12 30	N/C N/C

I/O Type:	Single Ended 5v Inputs / Outputs
Circuit Type:	Sink (4.7K ohm pull-up to internal +5v)
Input Impedance:	4.7K ohms to internal +5v
Maximum Input Voltage:	-1.0 v , +7.0v
Logic 0 Input Threshold:	+0.8 v max
Logic 1 Input Threshold :	+2.4 v min
Input Filtering:	10 microseconds typical
Output Sink Current	10 ma max
On State Output Voltage	+0.5v at 10 ma
Strobe Response:	Minimum Pulse Width: 25 microseconds Position Capture Delay: 10 to 280 microseconds
Notes:	For digital servos, these points act as the PWM / ENBL outputs and Alarm inputs. For Aux Axis 3, these points are input only. The listed 0v pins should be normally used for the signal return.



5v Differential Outputs

Circuit Identifier	*Digital Servo Axis 1, 2 Circuit Function*	*Analog Servo Axis 1,2 and Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
OUT2	Ser Encoder Req (+) Ser Encoder Req (-)	Not Used Not Used	Not Used Not Used	OUT2P_A OUT2M_A	13 31	13 31	N/C N/C
OUT3	Faceplate 5v Output (+) Faceplate 5v Output (-)	Faceplate 5v Output (+) Faceplate 5v Output (-)	Faceplate 5v Output (+) Faceplate 5v Output (-)	OUT3P_A OUT3M_A	14 32	14 32	5 13

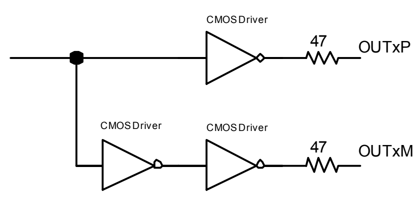
I/O Type: 5v Differential Outputs

Circuit Type: Differential Totem Pole (Source / Sink)

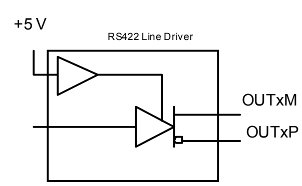
Output Source/Sink Current: 20 ma max

Output Voltage: +/- 1.5 v min across 120 ohm differential load

Notes: Servo Axis 1 and Aux Axis 3 use CMOS Drivers with 47 ohm series resistors, Servo Axis 2 and Aux Axis 4 use RS-422 Line Drivers.



Axis 1 and Aux Axis 3



Axis 2 and Aux Axis 4

24v DC Optically Isolated Output

Circuit Identifier	*Servo Axis 1, 2 Circuit Function*	*Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
OUT1	Faceplate 24v Output (+)	Faceplate 24v Output (+)	Faceplate 24v Output (+)	OUT1P_A	18	18	8
	Faceplate 24v Output (-)	Faceplate 24v Output (-)	Faceplate 24v Output (-)	OUT1M_A	36	36	16

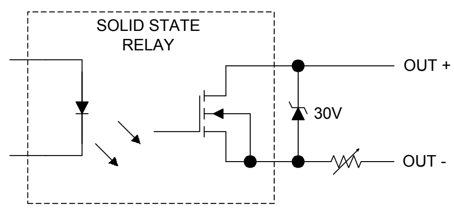
I/O Type: 24v DC Optically Isolated Output

Circuit Type: Isolated Solid State Relay (SSR)

Output Current: 125 ma continuous, 500 ma for 10 ms (resistive or inductive)

Output Voltage Drop: 1.0 v max at 0.125 amps

Notes: Output is protected by a 30v transzorb and a 0.2 amp Polyswitch. If a short circuit occurs, the output will automatically switch to a high impedance state until the load is removed. The load should not be re-applied for 60 seconds. This is a dc output and it will appear to be always ON if connections to it are reversed.



Optically Isolated Enable Relay Output

Circuit Identifier	*Servo Axis 1, 2 Circuit Function*	*Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
ENBL	Servo MCON (+)	Not Used	Not Used	ENBL1_A	15	15	N/C
	Servo MCON 0v	Not Used	Not Used	ENBL2_A	33	33	N/C

I/O Type: Optically Isolated Enable Relay Output

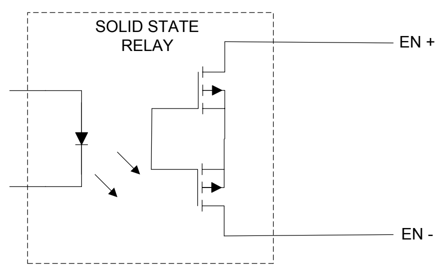
Circuit Type: Isolated AC Solid State Relay (SSR)

Output Current: 30 ma continuous, 50 ma for 10 ms

Output Voltage Drop: 1.0 v max at 10 ma

Notes: This is a low current SSR output. The output is ON when the associated faceplate Axis Enabled LED is illuminated. This occurs when:

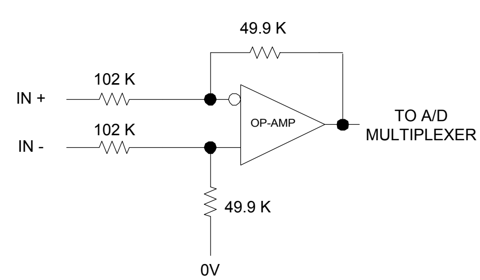
- The servo is enabled for Axis 1 or 2
- A *Force Digital Servo Velocity* AQ Cmd is used on Axis 1 or 2
- A *Force Analog Output* AQ Cmd is used on Axis 3 or 4



Differential +/- 10v Analog Inputs

Circuit Identifier	*Digital Servo Axis 1, 2 Circuit Function*	*Analog Servo Axis 1, 2 and Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
AIN1	IR Phase Current (+) IR Phase Current (-)	Faceplate Analog In (+) Faceplate Analog In (-)	Faceplate Analog In (+) Faceplate Analog In (-)	AIN1P_A AIN1M_A	7 25	7 25	N/C N/C
AIN2	IS Phase Current (+) IS Phase Current (-)	Faceplate Analog In (+) Faceplate Analog In (-)	Faceplate Analog In (+) Faceplate Analog In (-)	AIN2P_A AIN2M_A	8 26	8 26	N/C N/C

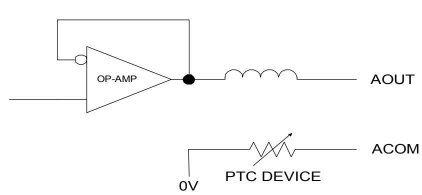
I/O Type:	Differential +/- 10v Analog Inputs
Circuit Type:	Differential Input
Input Impedance:	102K ohms common mode with respect to faceplate connector 0v 204K ohms differential
Maximum Input Voltage:	+/- 15 v common mode with respect to faceplate connector 0v +/- 20 v differential
Resolution:	15 bits
Linearity:	13 bits
Input Offset:	+/- 1000 uv
Gain Factor:	+/- 10.0v = +/- 32,000 counts
Gain Accuracy:	+/- 0.5 %
Update Rate:	1. 10 milliseconds + PLC sweep time when data is reported to PLC %AI table. 2. 2 milliseconds when used as Analog Master Input in Follower mode.
Notes:	Use faceplate 0v pins for common mode reference.



Single Ended +/- 10v Analog Output

Circuit Identifier	*Analog Servo Axis 1, 2 Circuit Function*	*Digital Servo Axis 1, 2 and Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
AOUT1	Analog Servo Velocity Command	Faceplate Analog Out	Faceplate Analog Out	AOUT_A	6	6	4
ACOM	Analog Out Com	Analog Out Com	Analog Out Com	ACOM_A	24	24	12

I/O Type:	Single Ended Analog Output
Circuit Type:	Op Amp Voltage Follower Output
Load Impedance:	2K ohms minimum
Output Current:	5 ma max
Resolution:	13 bits
Linearity:	13 bits
Output Offset Voltage:	+/- 500 uv max
Force D/A Gain Factor:	+/- 10.0v = +/- 32000 counts
Gain Accuracy:	+/- 1.0 %
Force Analog Output Update Rate:	1. PLC sweep rate when used by *Force Analog Output* %AQ command. 2. 250 microseconds when used as Digital Servo tuning output.
Notes:	Since this is a single ended output, it should normally drive a user device with a differential input to prevent common mode noise problems. The positive differential input should be connected to AOUT and the negative differential input to ACOM. The *Select Analog Output Mode* %AQ command can be used to select the source for the analog output. Refer to Chapter 5 for more information.



+5v Power

Circuit Identifier	*Servo Axis 1, 2 Circuit Function*	*Aux Axis 3 Circuit Function*	*Aux Axis 4 Circuit Function*	*Signal Name (Axis 1 listed)*	*Faceplate Pin*	*Auxiliary Terminal Board*	*Servo Terminal Board*
P5V	5v Power	5v Power	5v Power	P5V_A	4	4	3
0V	0v	0v	0v	0V_A	22	22	11

I/O Type:	+5V Encoder Power
Circuit Type:	+5V Power with Electronic Short Circuit Protection
Output Voltage:	4.70 v to 5.20 v at 0.5 amp
Output Current:	0.5 amp max (total for all connectors)
Notes:	This output is intended to power external devices such as Incremental Quadrature Encoders requiring less than 0.5 amps total from all 4 axis connectors. The output current is provided by the PLC backplane +5v supply and is protected by an electronic short circuit protector in the DSM302 module. The listed 0v pin should normally be used as the power return signal.

Chapter 4 Configuring the DSM302

This chapter describes all configuration details necessary to set up the DSM302 for a specific application. Refer to Chapter 2 for start-up instructions on how to configure the system to send a *Jog* command to the DSM in order to test that system components are operable.

The DSM302 module is configured using the Logicmaster 90-30 configuration software. Configuration is a two-part procedure consisting of:

- Rack Slot Configuration
- Module Configuration

Note

In order to configure the DSM302 module you must have version 8.02, or later, of the Logicmaster 90-30 configuration software. Certain functions, i.e. follower mode ratios in excess of 1:32 require Logicmaster 9.01 or later. Also, Version 1.0 of the GE Fanuc VersaPro graphical PLC programming software supports the DSM302.

Rack/Slot Configuration

Logicmaster 90-30 is used to define the type and location of all modules present in the PLC racks. This is done by first completing setup screens that represent the modules in a baseplate, then saving the information to a configuration file, which is then downloaded to the PLC CPU.

Refer to Chapter 2, Section 4 for the details of selecting a DSM302 baseplate and slot location in the PLC rack. After the Rack Slot Configuration is defined, proceed to the second part of the configuration process, Module Configuration, where you will configure the DSM302 for your specific application requirements.

Module Configuration

Setting the Configuration Parameters

As with I/O Rack Configuration, module configuration is done by completing screens in the Logicmaster 90-30 configuration software.

DSM302 configuration data consists of four types:

- Module Configuration Data
- Serial Communications (Programmer Port) Configuration Data
- Axis Configuration Data
- Program Zero

For additional details concerning the operation of DSM302 Configuration or Programming software packages, please consult the appropriate User Manuals for those particular software products. See "Related Publications" in the Preface of this Manual.

Module Configuration Data

During each CPU sweep certain data is automatically transferred both ways between the DSM302 and the CPU. CPU Interface data references the starting locations for the automatic transfers. The configuration parameters in Module Configuration Data are described in Table 4-1. All *Reference Section* designations shown in the tables pertain to this chapter.

Table 4-1. Module Configuration Data

Configuration Parameter	Description	Values	Logicmaster Defaults	Units	Reference Section
Ref Adr	Start address for %I ref type (64 bits)	CPU Dependent	%I00001 or next higher reference	N/A	—
Ref Adr	Start address for %Q ref type (64 bits)	CPU Dependent	%Q00001 or next higher reference	N/A	—
Ref Adr	Start address for %AI ref type (40 words for typical standard mode applications; 50 words for typical follower mode applications; 64 words for special applications)	CPU Dependent	%AI00001 or next higher reference	N/A	—
Ref Adr	Start address for %AQ ref type (6 words for typical standard mode applications; 9 words for typical follower mode applications; 12 words for special applications)	CPU Dependent	%AQ00001 or next higher reference	N/A	—
AI / AQ Len	Selects appropriate length combinations of the %AI and %AQ reference addresses (40/6 for typical standard mode, 50/9 for typical follower mode and 64/12 for special applications)	40 / 6 50 / 9 64 / 12	40 / 6	N/A	1.01

Table 4-1. Continued – Module Configuration Data

Configuration Parameter	Description	Values	Logicmaster Defaults	Units	Reference Section
Fdback Type	Feedback Type	ENCODER LINEAR[2] RESOLVR[1] CUSTOM 1[1] CUSTOM 2[2] DIGITAL	ENCODER	N/A	1.02
Ctl Loop	Control Loop Type	STANDARD FOLLOWER CCL1[1] CCL2[2]	STANDARD	N/A	1.03
Servo Cmd	Servo Interface Type	ANALOG DIGITAL DUAL[1]	ANALOG	N/A	1.04
Motor1 Type Motor2 Type	GE Fanuc Motor Type	0 . . . 127 (0 or valid FANUC Motor types ONLY!)	0 (no motor - digital) (analog mode)	N/A	1.05
Motor1 Dir Motor2 Dir	Motor direction for positive velocity command	POS NEG	POS	N/A	1.06

[1] Reserved for future use; not implemented at this time
[2] For special purpose applications only

1.01 AI/AQ Len. Sets the number of %AI and %AQ references assigned to the DSM302. The possible combinations of %AI and %AQ references are *40/6*, *50/9* and *64/12*. 40/6 is typically selected for Standard mode applications. Follower mode applications typically use the 50/9 selection. The 64/12 selection is normally reserved for applications requiring additional analog I/O (i.e., to gain access to the analog I/O on module connectors **C** and **D**). (Default = *40/6*)

1.02 Fdback Type. *ENCODER* selects incremental quadrature encoder [A quad B (x4)] input mode. *LINEAR* selects linear transducer (absolute feedback) input mode (for special purpose use with CCL2). *RESOLVR* selects resolver (absolute feedback) input mode. *CUSTOM1* and *CUSTOM2* configure the DSM302 inputs for special applications. *DIGITAL* selects GE Fanuc Digital AC servo encoder input mode. If DIGITAL is selected, the ***Servo Cmd*** configuration parameter must also be set to DIGITAL. (LM90 Default = *ENCODER*).

Note

Normally, only *DIGITAL* or *ENCODER* feedback option should be selected. The *RESOLVR, CUSTOM1, and CUSTOM2* options are not currently supported. The *LINEAR* option is reserved for special applications in conjunction with CCL2.

1.03 Ctl Loop. *STANDARD* selects the normal DSM302 motion control loop. The STANDARD loop provides a velocity command (Analog mode) or velocity/position command (Digital mode) proportional to position error, with optional Velocity Feed forward. *FOLLOWER* selects a control loop that allows ratio tracking of a master input. *CCL1* and *CCL2* (Custom Control Loops) are individually designed for special applications and should not be selected. (Default = *STANDARD*).

1.04 Servo Cmd. This parameter defines the type of command output provided to the servo sub-system. *DIGITAL* selects a special digital output for GE Fanuc Digital servo drives. *ANALOG* selects a +-10 volt velocity command for standard analog servo drives. *DUAL* is not supported. (LM90 Default = *ANALOG*).

1.05 **Motor Type**. Selects the type of FANUC AC servomotor to be used with the DSM302 in Digital Mode ONLY. The DSM302 internally stores setup motor parameter tables for each of the FANUC motors supported. A particular motor for the indicated axis is selected in the Logicmaster 90-30 configuration fields ***Motor1 Type*** or ***Motor2 Type***. A motor type of *0* for a particular axis disables digital servo control by the DSM302 for the digital servo axis. **Motor type must be set to zero when no digital servo is attached if any %Q bit commands or %AQ data commands will be sent to the axis.** Supported FANUC Motor types are listed in the tables below.

The Motor Type must be 0 for ANALOG Mode or if no motor is attached to the axis. Default: 0.

FANUC Motor part numbers are used to determine the proper FANUC Motor type code and are in the form A06B-xxxx-yyyy, where xxxx represents the motor specification field. For example: When reading a motor number from a motor label of A06B-0032-B078, the motor specification digits 0032 indicate the motor model of β2/3000. The β Series table references the *Motor Type Code* (36) needed for the configuration field. Supported FANUC Motor types are listed in the tables below. The list of supported motors may be expanded in future releases.

α Series FANUC Servo Motor

Motor Type Code	Motor Model	Motor Specification
61	α 1/3000	0371
46	α 2/2000	0372
62	α 2/3000	0373
15	α 3/3000	0123
16	α 6/2000	0127
17	α 6/3000	0128
18	α 12/2000	0142
19	α 12/3000	0143
27	α 22/1500	0146
20	α 22/2000	0147
21	α 22/3000	0148
28	α 30/1200	0151
22	α 30/2000	0152
23	α 30/3000	0153
30	α 40/2000	0157
29	α 40/FAN	0158

α L Series FANUC Servo Motor

Motor Type Code	Motor Model	Motor Specification
56	α L3/3000	0561
57	α L6/3000	0562
58	α L9/3000	0564
59	α L25/3000	0571
60	α L50/2000	0572

α C Series FANUC Servo Motor

Motor Type Code	Motor Model	Motor Specification
7	α C3/2000	0121
8	α C6/2000	0126
9	α C12/2000	0141
10	α C22/1500	0145

α HV Series FANUC Servo Motor

Motor Type Code	Motor Model	Motor Specification
3	α 12HV/3000	0176
4	α 22HV/3000	0177
5	α 30HV/3000	0178

DSM firmware revision 1.10 or later is required for α HV Series Motors

α M Series FANUC Servo Motor

Motor Type Code	Motor Model	Motor Specification
24	α M3/3000	0161
25	α M6/3000	0162
26	α M9/3000	0163

β Series FANUC Servo Motor

Motor Type Code	Motor Model	Motor Specification
13	β 0.5/3000	0013
35	β 1/3000	0031
36	β 2/3000	0032
33	β 3/3000	0033
34	β 6/2000	0034

1.06 **Motor Dir**. For all GE Fanuc digital servos, a configured motor direction of POS (Positive) defines the positive axis direction as counter clockwise (CCW) motor shaft rotation when viewed looking into the motor shaft. A configured Motor direction of NEG (Negative) defines the negative axis direction as clockwise (CW) shaft rotation.

For analog servos, a configured analog motor direction of POS (Positive) defines the positive axis direction as encoder channel A leading channel B. A configured analog motor direction of NEG (Negative) defines the negative axis direction as encoder channel B leading channel A. In practice, the motor direction configuration allows the user to easily reverse the motion caused by all commands without having to change the motion program.

Serial Communications Port Configuration Data

Motion programs (in addition to Program Zero) for the DSM302 can be programmed using the APM Motion Programmer software. The computer running the Motion Programmer software connects to the Serial Communications Port (which supports the SNP protocol) on the DSM302 faceplate. The DSM's Serial Communications Port is labeled "COMM" on the module's faceplate.

The DSM302's Serial Communications Port must be configured properly to communicate with the Motion Programmer (PC). Make sure the Motion Programmer configuration parameters and the DSM302's Serial Communications Port configuration parameters match. The DSM302's Serial Communications Port configuration parameters are described in Table 4-2.

Table 4-2. Serial Communications Port Configuration Data

Configuration Parameter	Description	Values	Defaults	Units
Baud Rate	Baud rate of SNP Port	300, 600, 1200, 2400, 4800, 9600, 19200	19200	N/A
Parity	Parity	ODD, EVEN, NONE	ODD	N/A
Stop Bits	Number of stop bits	1 or 2	1	N/A
Data Bits	Number of data bits	7 or 8	8	N/A
Modem TT	Modem turnaround time	0…255	0	1/100 sec.
Idle Time	Maximum link idle time	1...60	10	sec
SNP ID	SNP ID	6 characters consisting of A-F and 0-9. First character must be A-F	A00001	N/A

Baud Rate. The baud rate parameter specifies the transmission rate, in bits per second, of data through the serial port.

Parity. Specifies whether or not a parity bit is to be used (**NONE** if not), and if so, whether it should be **ODD** or **EVEN**.

Stop Bits. All serial communications devices use at least one (**1**) stop bit. For slower devices, set this parameter to two (**2**) stop bits.

Data Bits. Specifies whether seven (**7**) or eight (**8**) data bits are to be used to represent a single "piece" of data. Typically seven data bits are used for ASCII-only data and eight where binary data is to be transmitted.

Modem TT. When utilizing a modem, a Modem Turnaround Time must be specified. This is the time required for the modem to start data transmission after receiving the transmit request. If no modem is used, 0 should be specified. If a modem is used, a value greater than 0 must be specified.

Idle Time. Specifies the time, in seconds, that the DSM302 will wait for a new message to be received from the master device (for example, Motion Programmer or HMI) before assuming that communications have been lost or terminated. In such a case, the DSM302 will reinitialize to wait for the start of a new SNP connection sequence.

SNP ID. An identifier consisting of from 0 to 6 characters consisting of A-F and 0-9. The first character specified must be in the set A-F. An SNP ID need not be specified if the programmer is directly connected to a single module. The identifier must be utilized for a multi-drop network. The DSM302 will support multi-drop connections only if the RS232 connection is converted to RS422/485.

Axis Configuration Data

The DSM302 configuration parameters define such things as control mode, motor type, feedback type, as well as tuning settings that match the servo system to the physical equipment being driven. Therefore, most configuration parameters values do not change, and so they can not be accessed in the motion program. However, a relatively small number of configuration parameters may require changing to accommodate changes in the user's application. Examples of these parameters are, In Pos Zone, Jog Vel, and Jog Acc. Changes to these parameters are supported by PLC %AQ Immediate Commands. Refer to Chapter 5, Section 4 for more information. The configuration parameters for each control loop mode are defined and briefly described here. The numbers in the "Ref" column refer to item numbers in this chapter.

Table 4-3. Axis Configuration Data

Configuration Parameter	Description	Values	Logicmaster Defaults	Units	Valid Modes	Ref
User Units	User Units Value	1...65,535 (Standard Mode Only)	1	N/A	Standard	2.01
Counts	Feedback Counts	1...65,535 (Standard Mode Only)	1	N/A	Standard	2.01
OT Limit Sw	Over travel Limit Switch Enable / Disable	ENABLED DISABLED	ENABLED	N/A	Standard Follower	2.02
Pos Err Lim	Position Error Limit	256...60,000 (See Note 1)	+ 4096	user units	Standard Follower	2.03
In Pos Zone	In Position Zone	0...255	10	user units	Standard Follower	2.04
Pos Loop TC	Position Loop Time Constant	0, 5...10,000 (See Note 1)	1000	ms	Standard Follower	2.05
Vel at 10 V	Velocity for 10 V Output	400...1,000,000 (See Note 1)	+4000	user units / sec	Standard Follower	2.06
Vel FF %	Velocity Feed Forward	0...100	0	%	Standard Follower	2.07
Intgr TC	Integrator Time Constant	0...10,000	0	ms	Standard Follower	2.08
Intgr Mode	Integrator Mode	OFF CONT IN ZONE	OFF	N/A	Standard Follower	2.09
Rev Comp	Reversal Compensation	0...255	0	user units	Standard	2.10
DisDly	Drive Disable Delay	0...65,535	100	ms	Standard Follower	2.11
Vel Lp Gain (Digital Mode only)	Velocity Loop Gain	0...255	16	N/A	Standard Follower	2.12
Fdback Mode (Digital Mode only)	Feedback Mode	INC ABSOLUTE	INC	N/A	Standard Follower	2.13
Jog Vel	Jog Velocity	1...8,388,607	+1000	user units / sec	Standard Follower	2.14
Jog Acc	Jog Acceleration	1...134,217,727	+10,000	user units/sec/sec	Standard Follower	2.15
Jog Acc Mod	Jog Acceleration Mode	LINEAR S-CURVE	LINEAR	N/A	Standard Follower	2.16

Table 4-3. - Continued - Axis Configuration Data

Configuration Parameter	Description	Values	Logicmaster Defaults	Units	Valid Modes	Ref
Axis Mode	Axis Mode	CONTINU LINEAR/ ROTARY1 (See Note 2) ROTARY2 (See Note 2)	CONTINU	N/A	Standard Follower	2.17
Tuning Par1	Tuning Parameter 1	0...63	0	N/A	Standard Follower	2.18
Tuning Dat1	Tuning Parameter 1 Data	-32,768...+32,767	0		Standard Follower	2.19
Tuning Par2	Tuning Parameter 2	0...255	0	N/A	Standard Follower	2.20
Tuning Dat2	Tuning Parameter 2 Data	-32,768...+32,767	0		Standard Follower	2.21
Hi Limit	High Count Limit	-8,388,608...+8,388,607	+8,388,607	user units	Standard Follower	2.22
Lo Limit	Low Count Limit	-8,388,608...+8,388,607	-8,388,608	user units	Standard Follower	2.23
Home Positn	Home Position	-8,388,608...+8,388,607	0	user units	Standard Follower	2.24
Home Offset	Home Offset Value	-32,768...+32,767	0	user units	Standard Follower	2.25
Fnl Hm Vel	Final Home Velocity	1...65,535	+500	user units / sec	Standard Follower	2.26
Find Hm Vel	Find Home Velocity	1...8,388,607	+2000	user units / sec	Standard Follower	2.27
Home Mode	Find Home Mode	HOMESW/MOVE+/ MOVE-	HOMESW	N/A	Standard Follower	2.28
Pos EOT	Positive End of Travel	-8,388,608...+8,388,607	+8,388,607	user units	Standard	2.29
Neg EOT	Negative End of Travel	-8,388,608...+8,388,607	- 8,388,608	user units	Standard	2.30
Ratio A Val	Follower A/B Ratio A	-32768...+32767 (not 0)	1	N/A	Follower	2.31
Ratio B Val	Follower A/B Ratio B	1...32767	1	N/A	Follower	2.31
Mstr Source	Master Axis Source	ENC3/INT ENC2 ANALOG	ENC3/INT	N/A	Follower	2.32
+Vlim	Master Velocity + Limit	0...1000	1000	counts / ms	Follower	2.33
-Vlim	Master Velocity - Limit	-1000...0	-1000	counts / ms	Follower	2.34
FollwrEnInp	Follower Enable Input	0...16	0	N/A	Follower	2.35
Follwr Mode	Follower Mode	NORMAL ACC RAMP WINDER WINDER L WINDER R CAM 1 (See Note 2) CAM 2 (See Note 2) CAM 3 (See Note 2)	NORMAL	N/A	Follower	2.36

Table 4-3. - Continued - Axis Configuration Data

Configuration Parameter	Description	Values	Logicmaster Defaults	Units	Valid Modes	Ref
MkupTime	Ramp Acc Makeup Time	0...32,000	0	ms	Follower	2.37
Winder Znlen	Winder Zone Length	100...+8,388,607	10,000	counts	Follower	2.38
Enc3 Hi Lim	Aux Axis 3 Position Hi Count Limit	-8,388,608...+8,388,607	+8,388,607	counts	Follower	2.39
Enc3 Lo Lim	Aux Axis 3 Position Low Count Limit	-8,388,608...+8,388,607	-8,388,608	counts	Follower	2.40
Master Home Positn	Aux Axis 3Home Position	-8,388,608...+8,388,607	0	counts	Follower	2.41
Master Home Switch	Aux Axis 3Home Switch Enable	ENABLED DISABLED	ENABLED	N/A	Follower	2.42
Anlg Max Vel	Analog Maximum Velocity	0...1,000,000	0	counts / sec	Follower	2.43

Note 1 Scaling-dependent configuration parameter whose value range depends upon user unit scaling. Values shown in table are for default scaling. Refer to the descriptions below for calculating the current values when using non-default scaling of ***User Units*** and ***Counts*** parameters.

Note 2 These parameters are reserved for future use.

2.01 User Units, Counts. The User Units to Counts ratio sets the number of programming units for each position feedback count. This allows the user to program the DSM302 in application-specific units. The User Units and Counts values must be within the range of 1 to 65,535. The User Units to Counts ratio must be within the range of 8:1 to 1:32. For example, if there is 1.000 inch of travel for 8192 feedback counts, a 1000:8192 User Units:Counts ratio sets 1 User Unit equal to 0.001 inch. Default is 1:1.

For **FOLLOWER** mode this ratio is fixed at one to one (1:1) and cannot be changed. The default User Unit is one encoder count.

For **STANDARD** mode, the User Units to Counts ratio sets the number of position programming units for each feedback count. It is a requirement to set this value correctly for the mechanical systems coupled to the axis, otherwise movement to unsafe and inaccurate positions may occur. **It is important to set this relationship at the beginning of the configuration session; most other configuration fields are specified in user units**. For example, Velocity will be specified in user units per second and Acceleration will be specified in user units per second per second.

The DSM300 Series module has a very powerful scaling feature. A User Unit to Counts ratio can be configured to allow programming in other than default counts. In a simplified example, suppose an encoder feedback application has an encoder that produces 1,000 counts per revolution (250 lines) and is geared to a machine that produces one inch per revolution. The default unit would be one thousandth of an inch per count. However, you may want to write programs and use the DSM300 Series module with metric units. A ratio of 2540 User Units to 1000 Counts can be configured to allow this. With this ratio, one user unit would represent .01 millimeters. 2540 user units would represent 25.40 millimeters (one inch) of travel.

The example below illustrates how to meet the requirements that the User Units and Counts values be within the range of 1 to 65,535, and the User Units to Counts ratio be within the range of 8:1 to 1:32.

The basic equation we need to satisfy is:

$$\frac{\text{User Units}}{\text{Counts}} = \frac{(\text{Load Movement Per Motor Rotation}) \div (\text{Desired Resolution})}{\text{Encoder Counts Per Motor Rotation}}$$

The numerator and denominator must each fit within the RANGE limits. The reduced fraction must fit between the RATIO limits. The decimal point is always implied, not used. The User Units to Counts ratio is always expressed as an integer ratio.

Example Application

Use the User Units to Counts ratio to configure the DSM302 so you can program in engineering units rather than encoder counts. As an example, assume a machine has a motor with a motor-mounted quadrature encoder connected through a gear reducer to a spur gear. The spur gear is mounted to the end of a pinch-roller shaft. The pinch roller feeds sheet material for a cut-to-length application. The motion program will specify the length of cut sheets. The programmer wishes to program in 0.01-inch resolution.

The following data is given:

- 2000 line encoder (x4 = 8000 counts per encoder revolution)
- 20:1 gear reduction
- 14.336 inch diameter spur gear
- inch desired programming unit

Although several approaches are possible, the most straightforward is to base the calculations on a single spur gear revolution.

First determine the number of User Units per spur gear revolution.

14.336 inch diameter * π (pi) = 45.0378 inches circumference

45.0378 inches / 0.01 inch desired programming units = 4503.78 User Units per revolution of spur gear

Then determine the number of encoder counts per spur gear revolution.

2000 lines * 4 * 20 = 160,000 encoder counts per spur gear revolution

The * 4 results from the quadrature encoder generating 4 counts per line
The * 20 results from the 20:1 gear reduction

Then check the value of the User Units to Counts ratio. The ratio must be in the 8:1 to 1:32 range and the two numbers must be in the 1 to 65535 range.

4503.78 User Units / 160,000 encoder counts = 0.02815 or 1:35.5

This ratio is too small, so something must be changed. Any of the following system components could be changed to solve the problem:

- The spur gear diameter to 15.92 inch or larger
- The encoder lines per revolution to 1800 or less
- The gear reduction to 18:1 or less
- The desired programming unit to 0.001 inch

By far the easiest component to change is the desired programming unit to 0.001 inch.

Now recalculate to determine the revised User Units per revolution using 0.001 inch programming unit.

14.336 inches diameter * pi = 45.0378 inches circumference

45.0378 inches / 0.001 inch programming unit = 45,037.8 User Units per revolution of spur gear

Thus, the User Units to Counts ratio is 45,038 / 160,000 = 0.2815 or about 1:3.6 which is within the valid ratio range.

So a 45,038 / 160,000 ratio would be used except that 160,000 is larger than the maximum 65,535 range value. Dividing both numbers by 10 solves this to make the ratio 4,504 / 16,000. Note that in the above example, we simply reduced the fraction and ignored the slight rounding error

One method of avoiding “rounding” is to express the numeric ratio as a fraction. From the previous example, any number set that produced a 0.2815 ratio could be used. An example is 2815 / 10000.

Another approach is to rationalize the fraction (reduce it to its lowest terms). This is done by evenly dividing both the numerator and denominator by successively smaller prime numbers, beginning with the largest prime that will evenly divide into both the numerator and the denominator, until no more division without remainders is possible.

Always maintain an exact integer fraction, a decimal ratio expressed as a fraction or a rationalized fraction when configuring the User Units to Counts ratio for the best accuracy. The user must determine if the rounding error, if present, is of significance. A rotary mode application that always operates in one direction will accumulate rounding errors over time and “drift”. A linear application will only accumulate error for the length of travel then “rewind” as the axis reverses.

2.02 OT Limit Sw.

Selects whether the DSM300 Series module uses the hardware over travel limit switch inputs.

DISABLED, the limit switch inputs may be used as general-purpose motion program flow control and program branching inputs (i.e. CTL05-CTL08 bits).

ENABLED, indicates that the DSM300 will check the axis over travel inputs continuously, every 2 milliseconds whenever the %I *Drive Enabled* input is true. If either limit switch opens (the input goes to logic zero, Off) all motion is immediately commanded to stop. No deceleration control is active; the servo velocity command is set to zero. The solid state axis enable relay will not open until after the *%Q Enable Drive* command is set to zero. An error code indicating which limit is tripped is reported to the %AI Axis Error Code. At this point, only one DSM302 action is allowed: the appropriate %Q *Jog* and %Q *Clear Error* bits may be used simultaneously to back away from

the Limit Switch. The %Q *Clear Error* bit must be maintained ON to Jog off the limit switch. The user may also manually move the disabled axis off the limit switch. After the alarm is clear, normal operation may resume. Force D/A commands ignore the limit switches and should be used with caution.

2.03 Pos Err Lim. Position Error Limit (User Units).

The Position Error Limit is the maximum *Position Error (Commanded Position - Actual Position*) allowed when the DSM302 is controlling a servo. Position Error Limit should normally be set to a value 10% to 20% higher than the highest *Position Error* encountered under normal servo operation. Default: 4096.

The Position Error Limit range formula is:

$$256 \times (\text{user units/counts}) \leq \text{Position Error Limit} \leq 60{,}000 \times (\text{user units/counts})$$

If Velocity Feedforward is not used, Position Error Limit can be set to a value approximately 20% higher than the *Position Error* required to produce a 4000-rpm command. *The Position Error* (User Units) required to produce a 4000 rpm command with 0% Velocity Feed forward is:

$$\text{Position Error (user units)} = \frac{\text{Position Loop Time Constant (ms)} \times \text{Servo Velocity @ 4000 rpm (user units/sec)}}{1000}$$

Example

The user units:counts ratio is 2:1 and the Position Loop Time Constant is 50 ms.

Step 1:

$$\text{Calculate servo velocity at 4000 rpm} = \frac{(2 \text{ user units/count}) \times (8192 \text{ counts/rev}) \times (4000 \text{ revs/minute})}{(60 \text{ seconds/minute})}$$

$$= 1{,}092{,}266 \text{ user units/second}$$

Step 2:

$$\text{Calculate } \textit{Position Error} \text{ at 4000 rpm} = \frac{(50 \text{ milliseconds}) \times (1{,}092{,}266 \text{ user units/second})}{1000 \text{ milliseconds/second}}$$

$$= 54613 \text{ user units}$$

If Velocity Feedforward is used to reduce the following error, a smaller error limit value can be used, but in general, the error limit value should be 10% - 20% higher than the largest expected following error.

Note

An *Out of Sync* error will occur and cause a fast stop if the *Position Error Limit* Value is exceeded by more than 1000 counts. The DSM302 attempts to prevent an *Out of Sync* error by temporarily halting the internal command generator whenever position error exceeds the *Position Error Limit*. Halting the command generator allows the position feedback to catch up and reduce position error below the error limit value.

If the feedback does not catch up and the position error continues to grow, the *Out of Sync* condition will occur. Possible causes are:

1. Erroneous feedback wiring

2. Feedback device coupling slippage
3. Servo drives failure.
4. Mechanically forcing the motor/encoder shaft past the servo torque capability.
5. Commanded motor acceleration or motor deceleration that is greater than system capability.

2.04 In Pos Zone. In Position Zone (User Units). When the *Moving* %I bit is OFF and *Position Error* is less than or equal to the active ***In Position Zone*** value, the *In Zone* %I bit will be ON. This condition occurs at the end of each *Positioning Move* (PMOVE) command or any time the axis commands are halted and *Actual Position* has caught up to *Commanded Position* (e.g. for *Wait*, *Dwell*, *Feedhold*, or *Feedrate* Override % = 0).

In Follower mode, *In Zone* is ON under the same conditions described above. If the *Follower Enabled* %I bit is ON and the slave axis is following a master axis input, In *Zone* will be controlled only by the *Position Error* value as long as *Moving* is OFF. If *Moving* is ON due to a superimposed *Jog*, *Move at Velocity* or *Execute Program* command, *In Zone* will always be off. Default: 10.

2.05 Pos Loop TC. Position Loop Time Constant (milliseconds). The desired servo position loop time constant. This value configures the amount of time required for the servo velocity output to reach 63% of its final value when a step change occurs in the *Velocity* command. The lower the value, the faster the system response. Values that are too low will cause system instability and oscillation. Default: 1000.

Note

For accurate commanded velocity profile tracking, *Pos Loop TC* should be 1/4 to 1/2 of the MINIMUM system acceleration or deceleration time. For example if the fastest acceleration that must occur occupies 100msec of time the *Pos Loop TC* should be between 25 to 50msec. To maintain system stability, use the largest value possible.

For users familiar with servo bandwidth expressed in rad/sec:

Bandwidth (rad/sec) = 1000 / Position Loop Time Constant (ms)

For users familiar with servo gain expressed in ipm/mil:

Gain (ipm/mil) = 60 / Position Loop Time Constant (ms)

Table 4-4, Gain / Bandwidth / Position Loop Time Constant

Gain (ipm/mil)	Bandwidth (rad/sec)	Position Loop Time Constant (ms)
0.5	8.5	120
0.75	12.5	80
1.0	16.6	60
1.5	25.1	40
2.0	33.4	30
2.5	41.8	24
3.0	50	20

For applications that do not require feedback control or employ very crude positioning systems, an ***Open Loop Mode*** exists. Setting a zero Position Loop Time Constant, which indicates that the

positioning loop is disabled, selects this mode. Note that in Open Loop Mode, the only way to generate motion is to program a non-zero Velocity Feedforward. The *Position Error* is no longer used to generate motion because *Position Error* is based on position feedback and Open Loop Mode ignores all feedback.

Caution

The *Position Loop Time Constant* will not be accurate unless the *Vel at 10 V* value is set correctly.

2.06 Vel at 10 V. (User Units/Second.) All DSM302 and servo functions depend on this value being correct for proper operation.

For Digital Mode, the ***Velocity at 10 Volts*** *configuration* field should be set to a conversion constant value of <u>139820 multiplied by the decimal value of the ***User Units*** to ***Counts*** ratio</u>. For example: with a *User Unit* value of 1 and a *Counts* value of 2 the decimal value of their ratio would be 0.5. The conversion constant multiplied by 0.5 yields the value 69910 for the ***Vel at 10 V*** *(0.5 * 139820 = 69910).*

For Analog Mode, the ***Velocity at 10 Volts*** configuration field is the Actual Servo Velocity (User Units/second) desired for a 10 Volt DSM302 analog velocity command output to the servo. The DMS302 *Force D/A Output* %AQ Immediate Command and the *Actual Velocity* %AI status word can be used for a command voltage to empirically determine the proper configuration value if necessary.

The allowable ***Vel at 10V*** range for is:

(400 User Units/sec ≤ ***Vel at 10V*** ≤ 1,000,000 User units/sec)

In Digital Mode only, if the user sends the DSM302 a velocity command that exceeds the servo system capability, the DSM302 will *clamp* that command value at the appropriate maximum motor velocity boundary. **Note that no error will be reported back to the DSM302**.

See Appendix D, “Start-up and Tuning GE Fanuc Digital and Analog Servo Systems,” for more information on determining the correct value.

Default: <u>4000</u>.

Caution

The *Velocity at 10V* must be configured correctly in order for the *Pos Loop TC* and *VEL FF%* factors to be accurate.

2.07 Vel FF %. Velocity Feed forward gain (percent). The *Commanded Velocity* percentage that is added to the DSM302’s position loop velocity command output. Increasing Velocity Feedforward causes the servo to operate with faster response and reduced position error. The optimum value for each system has to be determined individually. For Digital Servos, a 95 to 100% ***Vel FF%*** value is a good starting point. For analog servos, 90-100 % is a typical value. If ***Vel FF%*** is changed, ***Pos Err Lim*** may require adjustment. Default: <u>0.</u>

Caution

The *Vel FF%* will not be accurate unless the *Vel at 10 V* value is first set correctly.

2.08 Intgr TC. Integrator Time Constant (milliseconds). This is the position loop position error integrator time constant. This value indicates the time required to reduce the position error by 63%. For example, if the Integrator Time Constant is 1000 (1 second), the *Position Error* would be reduced to 37% of its initial value after 1 second. A value of zero turns off the integrator. **If used, the Integrator Time Constant should be 5 to 10 times greater than the *Position Loop Time Constant* to prevent instability and oscillation.** Default: 0.

2.09 Intgr Mode. Integrator Mode. Position loop position error integrator operating mode. OFF means the integrator is not used. CONTINU means the integrator runs continuously even during servo motion. IN ZONE means the integrator only runs when the *Moving* %I status bit is OFF. ***Intgr Mode*** should normally be set to OFF. Continuous (CONTINU) may be used for Follower mode operation only when a constant or slowly changing master velocity is expected. This parameter should not be used to dampen disturbances in the position loop feedback. Never select continuous (CONTINU) for Standard mode applications. Default: OFF.

2.10 Rev Comp. Reversal Compensation (User Units). A compensation factor that allows the servo to reverse direction and still provide accurate positioning in systems exhibiting backlash. Backlash is exhibited by a servomotor that must move a small amount (lost motion) before the load begins moving when direction is reversed. For example, consider a dead bolt door lock. Imagine the servo controls the key in the lock and the feedback reports bolt movement. When the servo turns the key counterclockwise, the bolt moves left. However, as the servo turns the key clockwise, the bolt does not move until the key turns to a certain point. The Reversal Compensation feature adds in the necessary lost motion to quickly move the servo to where motion will begin on the feedback device. The DSM302 removes the compensation distance when a move in the negative direction is commanded, and adds the compensation distance before a move in the positive direction. The *%AI Actual Position* status word will indicate the actual distance moved *including* the reverse compensation distance. The machine actual position is therefore equal to the %AI Actual Position – Reversal Compensation distance after moves in the positive direction and immediately at start up. The actual value required for a particular application must be empirically determined. Default: 0.

Note

Reversal compensation applies only to the Standard mode of operation; it cannot be used in Follower mode.

2.11 DisDly. Servo Drive Disable Delay (milliseconds). The time delay from zero velocity command to the drive enable (digital servo MCON) signal switching off. Disable Delay is effective when the *Enable Drive* %Q bit is turned off or certain error conditions (Stop Mode) occur. Disable Delay should be longer than the deceleration time of the servo from maximum speed. Because turning OFF the *Enable Drive* %Q bit stops the DSM302 from commanding the servo, there are times when the drive enable signal should stay ON. For example, if the servo runs into an End of Travel Limit and the drive enable signal was immediately turned OFF due to the error, the servo may continue moving until it coasted to a stop. Thus, to allow the DSM302 to command and control a fast stop, the Drive Disable Delay should be longer than the deceleration time of the servo from maximum speed.

The disable delay may be used to control when torque is removed from the motor shaft. Applications using an electro-mechanical brake may need time for the brake to engage prior to releasing servo torque. Default: 100.

2.12 Vel Lp Gain. Velocity Loop Gain. Used to set velocity loop gain. This applies to GE Fanuc Digital Servos only. **This parameter is not used for Analog Servo Mode.** The formula

$$\text{Velocity Loop Gain} = \frac{\text{Load Inertia } (J_L) * 16}{\text{Motor Inertia } (J_M)}$$

can be used to select an initial velocity loop gain value. The allowable value range is 0 to 255. The value of 0 should be used if the motor shaft is not attached to a load.

Default: 16 (load inertia equals motor inertia).

2.13 Fdback Mode. Feedback Mode. Only used in Digital Mode. Used to configure Incremental or Absolute feedback type for the serial encoder. INC means the serial encoder is being used as an incremental encoder and encoder battery alarms will not be reported. ABSOLUTE means the serial encoder is being used as an absolute encoder (encoder backup battery installed) which maintains position if system power is cycled. In ABSOLUTE mode, encoder battery alarms will be reported. See appendix C, *Position Feedback Devices*, for more information. **This parameter is not used for Analog Servo Mode.** Default: INC.

2.14 Jog Vel. Jog Velocity (User Units/second). The velocity at which the servo moves during a *Jog*, *Find Home*, *Move at Velocity, Follower Ramp Control,* and *Abort* operation. The *Jog Velocity* is the default value used by motion programs when there is no programmed velocity. Default: 1000.

2.15 Jog Acc. Jog Acceleration Rate (User Units/second/second). The acceleration rate used during *Jog*, *Find Home*, *Move at Velocity*, *Follower Ramp Control*, and *Abort* operations. The *Jog Acceleration* is the value used by motion programs when there is no programmed acceleration. Default: 10000.

Note: A minimum value after scaling is used in the DSM302. This value is determined by the rule: Jog Acc * (user units/counts) >= 32 counts/sec/sec.

2.16 Jog Acc Mod. Jog Acceleration Mode (LINEAR or S-CURVE). The acceleration mode for *Jog*, *Find Home*, *Move at Velocity*, and *Abort* operations. LINEAR (constant acceleration) causes commanded velocity to change linearly with time. S-CURVE (jerk limited acceleration) causes commanded velocity to change more slowly than the linear mode at the beginning and end of acceleration intervals. **Motions using S-Curve acceleration require twice the time and distance to change velocity compared to motions using the same acceleration value with Linear acceleration.** In order to maintain equal machine cycle times, an S-Curve motion profile requires an acceleration value (and peak motor torque) twice as large as the equivalent Linear acceleration motion profile. Therefore a tradeoff between motor cost and machine cycle time may be necessary. Default: LINEAR.

2.17 Axis Mode. Axis Mode (CONTINU, LINEAR, ROTARY1 or ROTARY2). Sets the axis operation mode. CONTINU means that the end of travel (+EOT/-EOT) limits are not used and continuous programmed motion can be achieved using a series of incremental move commands. In this mode *Actual Position* will roll over at the configured HI/LO limits. LINEAR means that the +EOT/-EOT limits are used to restrict axis motion for both jogs and programmed moves. In LINEAR mode, the +EOT/-EOT limits must be less than or equal to the HI/LO rollover limits. If an EOT Limit is outside a HI/LO Limit, the DSM will internally set it equal to the HI/LO Limit. The ROTARY1 and ROTARY2 operation modes are reserved for the future. Default: CONTINU.

Note

When Follower mode is selected, the DSM302 internally assigns ***Pos EOT*** to the HI Limit rollover value and ***Neg EOT*** to the Lo Limit rollover value. The LINEAR ***Axis Mode*** selection can only be used in Follower mode when the axis *Actual Position* does not reach the Hi/Lo limit rollover boundaries. Therefore, at present in follower mode, ***Axis Mode = Linear*** provides little functionality over ***Axis Mode = Continu***.

Tuning Parameters

Tuning Parameters 1 and 2 are generic parameters used to select pre-defined tuning parameter settings such as the **Enhanced Position Loop Resolution** parameter settings, described below.

Warning

To avoid tuning parameter value conflicts, Tuning Par1 and Tuning Par2 should not contain the same non-zero value.

2.18 Tuning Par1. Tuning Parameter 1. Specifies axis-tuning parameter to be configured. Valid range is 0-63. Default: 0.

2.19 Tuning Dat1. Tuning Parameter 1 Data. Provides the value for the axis tuning parameter selected by Tuning Par1. Valid range is -32768...+32767. Default: 0.

2.20 Tuning Par2. Tuning Parameter 2. Specifies axis-tuning parameter to be configured. Valid range is 0-255. Default: 0.

2.21 Tuning Dat2. Tuning Parameter 2 Data. Provides the value for the axis tuning parameter selected by Tuning Par2. Valid range is –32768...+32767. Default: 0.

Table 4-5, Available Tuning Parameter Selections

Tuning Par1 or Tuning Par2 Value	Tuning Dat1 or Tuning Dat2 Value	Description
0	0	Not used (default values)
1	0…3	Enhanced Position Loop Resolution for Digital Servos (see table below)

Enhanced Position Loop Resolution for Digital Servos

For digital servos only, enhanced position loop resolution, at the expense of maximum supported motor velocity, has been added to the product in firmware release 1.20. Prior to this release, a non-configurable position loop resolution of 8192 counts per encoder revolution was provided. The table below describes the various selections now supported, along with the maximum supported motor velocity for each setting. Note that the configuration data is specified by entering a value of "1" (to select parameter 1) in the "Tuning Par1" or "Tuning Par2" field of the Axis-1 screen for axis-1 or Axis-2 screen for axis-2. The appropriate resolution setting value (0..3) is then entered in the corresponding "Tuning Dat1" field or "Tuning Dat2" field in the Axis-1 screen for axis-1 or Axis-2 screen for axis-2.

Table 4-6, Enhanced Position Loop Resolution

Tuning Par1 or Tuning Par 2 Value	Tuning Dat1 or Tuning Dat2 Value	Encoder Resolution (Counts per Revolution)	Maximum Motor Velocity (Revolutions per Minute)
1	0	8192 cts/rev	4400 rpm[1,2]
1	1	16384 cts/rev	3662 rpm[2]
1	2	32768 cts/rev	1831 rpm
1	3	65536 cts/rev	915 rpm[3]

[1] Default Setting.
[2] Some motors are restricted to a lower maximum rpm rating.
[3] α Series motors and encoders only.

2.22 Hi Limit. High Count Limit (User Units). When moving in the positive direction, the Actual Position will roll over to the low limit when this value is reached. **The Count Limits can be used for continuous rotary applications when the Axis Mode configuration is set to CONTINU.** The High Count Limit should always be set one User Unit smaller than the desired cycle. For example, a 360° machine would have a High Count Limit setting of 359. At the next count past 359, the count would roll over to the value set in the Low Count Limit parameter (0 in this example). See Appendix C for considerations when using an absolute mode encoder. Default: 8,388,607.

2.23 Lo Limit. Low Count Limit (User Units). When moving in the negative direction, the Actual Position will roll over to the high limit when this value is reached. **The Count Limits can be used for continuous rotary applications when the Axis Mode configuration is set to CONTINU.** See Appendix C for considerations when using an absolute mode encoder. Default: -8,388,608.

2.24 Home Positn. Home Position (User Units). The value assigned to *Commanded Position* when a *Find Home* cycle completes.

2.25 Home Offset. Home Position Offset (User Units). A value added to or subtracted from the servo's final stopping point when a *Find Home* cycle completes. **Home Offset** adjusts the final servo stopping point relative to the encoder marker. See chapter 6 for details on the home cycle. Default: 0.

2.26 Fnl Home Vel. Final Home Velocity (User Units/second). The velocity at which the servo seeks the final Home Switch transition and Encoder Marker pulse at the end of a Find Home cycle. This velocity is also used for the home cycle MOVE+ and MOVE- modes. See chapter 6 for details on the home cycle. Final Home Velocity must be slow enough to allow a 10 millisecond (filter time) delay between the final Home Switch transition and the Encoder Marker pulse. Default: 500.

2.27 Find Home Vel. Find Home Velocity (User Units/second). The velocity at which the servo seeks the initial Home Switch transitions during the *Find Home* cycle when the Home Mode is configured for HOMESW. If desired, Find Home Velocity can be set to a high value to allow the servo to quickly locate the Home Switch. Default: 2000.

2.28 Home Mode. Find Home Mode. The method used to find home during a *Find Home* cycle. HOME SWITCH indicates that a Home Switch is to be monitored to Find Home. MOVE+ and MOVE– specify direct positive and negative movement to the next encoder marker at the Final Home Velocity. See chapter 6, "Non-Programmed Motion," for details on the Home Cycle, Home Switch, Move+, and Move- Modes. Default: HOMESW.

2.29 Pos EOT. Positive Software End of Travel limit (User Units). If the DSM302 is programmed to go to a position greater than the Positive EOT, an error will result and the DSM302 will not allow axis motion. In Follower mode, the DSM302 internally sets ***Pos EOT*** to the ***HI Limit*** position rollover value (See Note under Axis Mode). **The Pos EOT limit applies only when the Axis Mode configuration is set to LINEAR.** Default: +8,388,607.

2.30 Neg EOT. Negative Software End of Travel limit (User Units). If the DSM302 is programmed to go to a position less than the Negative EOT, an error will result and the DSM302 will not allow axis motion. In Follower mode, the DSM302 internally sets ***Neg EOT*** to the ***LO Limit*** position rollover value (See Note under Axis Mode). **The Neg EOT limit applies only when the Axis Mode configuration is set to LINEAR.** Default: -8,388,608

If Pos / Neg EOT limits are both set to zero, the DSM302 uses +8,388,607 / -8,388,608 instead.

2.31 Ratio A and B Values. The A over B ratio sets the follower slave/master gear ratio.

$$\text{Follower Axis Motion (counts)} = \frac{\text{A * Master Reference Counts}}{\text{B}}$$

The range for A is –32,768 to +32,767 and B is 1 to +32,767. When A is negative, the slave axis will move in the opposite direction from the master. For DSM firmware revision 1.20 and later, the A/B slave/master follower ratio has been expanded from the original range of 32:1 to 1:32, now supporting 32:1 to 1:10,000. Starting with Logicmaster release 8.02, the %AQ Immediate Command, 2Dh, can also be used to specify this expanded range at runtime. However, specifying the expanded range (ratio greater than 1:32) in the module configuration requires Logicmaster 90-30 release 9.0 or later. Default: 1:1.

2.32 Mstr Source. Master Axis Source (ENC3/INT, ENC2 or ANALOG). The master reference for Servo Axis 1 can be selected as encoder 3/internal master (ENC3/INT), encoder 2 (ENC2), or the analog input (ANALOG). The master reference for Servo Axis 2 may be encoder 3/internal master or the analog input. Default: ENC3/INT.

2.33 +Vlim. Master Velocity + Limit (counts/ms). The maximum, positive, master velocity command in counts per millisecond allowed for the master axis input to the follower. If the master velocity command exceeds the Vlim value, the master velocity command will be clamped at the active Vlim value, an error (F2h) will be reported, and the Follower Velocity Limit %I Status Bit will be set.

The Master Velocity Limit value is used for an additional function if Follower Ramp Acceleration Control is active. The Follower Ramp Acceleration Control function has the capability to create a unique motion profile automatically that will be combined with the master command velocity. The calculated velocity used for this automatic profile generation will not exceed 80% of the Master Velocity Limit. The 80% limitation allows the remaining 20% of the Vlim value to be used to respond to changes in the master source velocity. See the description for Follower Make Up Time and Chapter 8, Follower Motion, for more information.

Superimposed motion (programmed motion executed while the follower is active i.e. Move Immediate, Jog, executing a motion program) is not affected by the Vlim value. It is possible when using superimposed moves with the follower axis enabled to exceed the velocity limit imposed on the follower motion. The Vlim status errors are returned only if the master command velocity exceed the active Vlim.

Configuring the +*Master Velocity Limit* to zero disables error reporting for the positive direction and provides unidirectional follower operation in the negative direction.

Default: +1000. Refer to Chapter 8, Follower Motion, for additional information.

2.34 -Vlim. Master Velocity – Limit (counts/ms). The maximum, negative, master velocity command in counts per millisecond allowed for the master axis input to the follower. If the master velocity command exceeds the Vlim value, the master velocity command will be clamped at the active Vlim value, an error (F2h) will be reported, and the Follower Velocity Limit %I Status Bit will be set.

The Master Velocity Limit value is used for an additional function if Follower Ramp Acceleration Control is active. The Follower Ramp Acceleration Control function has the capability to create a unique motion profile automatically that will be combined with the master command velocity. The calculated velocity used for this automatic profile generation will not exceed 80% of the Master Velocity Limit. The 80% limitation allows the remaining 20% of the Vlim value to be used to respond to changes in the master source velocity. See the description for Follower Make Up Time and Chapter 8, "Follower Motion", for more information.

Superimposed motion (programmed motion executed while the follower is active i.e. Move Immediate, Jog, executing a motion program) is not affected by the Vlim value. It is possible when using superimposed moves with the follower axis enabled to exceed the velocity limit imposed on the follower motion. The Vlim status errors are returned only if the master command velocity exceed the active Vlim.

Configuring the -*Master Velocity Limit* to zero disables error reporting for the negative direction and provides unidirectional follower operation in the positive direction.

Default: -1000. Refer to Chapter 8, Follower Motion, for additional information.

2.35 FollwrEnInp. Follower Enable Input. Selects the control bit, CTL01-CTL16, to be used as the Follower Enable input. The follower axis is enabled only when the selected input is ON **and** the *Enable Follower* %Q bit is also ON. When the *Enable Follower* %Qbit is ON, then the CTL bit chosen acts as a rising edge trigger to enable follower mode. After Follower is enabled, only the PLC *Enable Follower* %Q bit controls the active state of the following function. A value of 0 means the follower axis is enabled only by the *Enable Follower* %Q bit. CTL01-CTL08 and CTL13-CTL16 are faceplate inputs. *CTL09-CTL12* are %Q bits from the PLC. Default: <u>0</u>.

2.36 Follwr Mode. Follower Mode. Specifies the follower operating mode. NORMAL means the follower is enabled and disabled immediately (without any ramp acceleration control) by the *Enable Follower* %Q bit ANDed with the configured *Follower Enable Input* bit (CTL01-CTL16). ACC RAMP means the follower axis velocity is ramped at the active ***Jog Acceleration*** rate when the follower is enabled or disabled. The winder modes provide a follower cycle that reverses direction at each end of the configured ***Winder Zone Length***. If the ***Winder Zone Length*** is changed, WINDER provides equal taper, WINDER R provides taper of the right side only and WINDER L provides taper of the left side only. CAM1, CAM2 and CAM3 are reserved for future operating modes. Refer to Chapter 8 for additional details. Default: <u>NORMAL</u>.

2.37 MkupTime. Follower Acceleration Ramp Makeup Time (milliseconds). Specifies the time in milliseconds used to make up the master command counts lost during a follower acceleration ramp. If the distance correction is not possible in the makeup time (too small a value) then the correction time is longer and a warning error is reported. This setting only has an effect when the FOLLWR MODE is set to ACC RAMP.

If an acceleration ramp without any correction for lost counts is desired, Makeup Time should be set to 0.

Makeup time has a minimum value of 10, so for values entered in the range of 1…10 use 10 instead.

Default: 0.

Refer to Chapter 8, Follower Motion, Follower Axis Acceleration Ramp Control section, for a much more detailed discussion of this feature.

2.38 WinderZnlen. Winder Zone length. (Counts) Specifies the initial movement zone length for the follower winder motion. Refer to Chapter 8 for additional details of Winder operation. Default: 10000.

2.39 Enc3 Hi Lim. ENC3 (Aux Axis 3) High Count Limit (Counts). The maximum position which the Aux Axis 3 encoder will reach before the ***Actual Position*** %AI register rolls over to the value set in the ***Low Limit*** parameter. Default: +8,388,607

2.40 Enc3 Lo Lim. ENC3 (Aux Axis 3) Low Count Limit (Counts). The minimum position which the Aux Axis 3 encoder will reach before the ***Actual Position*** %AI register rolls over to the value in the ***High Limit*** parameter. Default: -8,388608

2.41 Master Home Position. Aux Axis 3 Home Position (Counts). The position to which Aux Axis 3 *Actual Position* is set at the completion of an Aux Axis 3 Find Home Cycle. Default: 0

2.42 Master Home Switch. Aux Axis 3 Home Switch Enable. Determines whether the Aux Axis 3 Home Switch is monitored during the Aux Axis 3 Find Home Cycle. ENABLED means the Home Switch input must be on for the Home Cycle to recognize the encoder marker pulse. DISABLED means the Home Switch input is ignored and the Aux Axis 3 Home Cycle will set ***Actual Position*** equal to ***Home Position*** when the first marker pulse occurs after the ***Find Home*** %Q bit command. Default: ENABLED

2.43 Anlg Max Vel. Analog Input Maximum Velocity (Counts/second). The follower master velocity that will be produced by 10.00V on the Analog Input. 0V always produces a velocity of 0 Counts per second. Although the Logicmaster 90 default is 0, when the DSM302 detects the default of zero, it internally sets the value to be equal to 32,000. Default: 0

Program Zero

For additional details concerning the operation of DSM302 Configuration or Programming software packages, please consult the appropriate User Manuals for those particular software products. See *Related Publications* in the Preface of this Manual.

Program Zero is a short default motion program (20 commands maximum in Standard mode; 9 commands maximum in Follower mode) which is defined in the Logicmaster 90-30 configuration software and downloaded to the DSM302 whenever the module is initialized by the PLC. Program Zero is programmed by entering motion commands in an English language text format similar to that of the MS-DOS based APM Motion Programmer software used for Programs 1-10. One main difference is that sub-routines are not allowed in Program Zero. The DSM302 determines whether Program Zero is an axis 1, axis 2, or multi-axis program according to which axis or axes are used in the program. For example, if all Program Zero commands contain Axis number = 1, Program Zero will be classified as a *single axis program* for Axis 1. Therefore Program Zero will be allowed to execute **concurrently** with another single axis program (1-10) for Axis 2. A *multi-axis program* 0 requires that **both** axes be home referenced prior to program execution. Only 1 multi-axis program may be active at a time.

```
PROGRAM 0   PAGE 1

Command       Data          Axis #
ACCEL         +0000000800   1
VELOC         +0000004000   1
PMOVE-AL      +0000010000   1
VELOC         +0000007500   1
PMOVE-AL      +0000000000   1
(NULL)        +0000000000   N/A
(NULL)        +0000000000   N/A
```

Figure 4-1. Example of a Logicmaster Program Zero

The Program Zero commands are entered using the Function keys (F1-F9). The cursor can be moved from field to field in Program Zero by using the Cursor Control (arrow) keys. Many Command fields have more than one choice. To choose, place the cursor on the desired Command field, then use the TAB key to toggle through the list of choices for that Command field. Paired with each command is a Data field for entering either a signed double integer (constant) or the number of a DSM302 data parameter (variable value), as appropriate for the configured command.

Logicmaster Program Zero Programmer Instruction Format

A Program Zero instruction line consists of a command and the associated data describing the command. The command and data are entered into areas on the program editor screen called fields. These fields are described below.

Command Field. The English language designation for the command. To enter a command, position the cursor on a command name field then press the desired function key (F1-F9). Most of the commands fall into groups such as CMOVE, PMOVE, VELOC, ACCEL, etc. The variations of these commands can be displayed (using the Tab key) after pressing the function key for the particular group.

For example, to program a **CMOVE-IS-P** (**C**ontinuous **MOVE**, **I**ncremental, **S**-curve acceleration, using a **P**arameter data value), first select a command field using the cursor control keys, then press the *CMOVE* function key. This causes CMOVE-AL to be displayed in the command field of the screen. With Logicmaster 90, the variations for the command options are cycled through by pressing the TAB key. For example, pressing the Tab key for a CMOVE-AL command cycles the command through CMOVE-AL-P, CMOVE-AS, etc. until the desired CMOVE-IS-P is displayed.

To program a JUMP command press F10 (MORE) and then press F1 (JUMP).

Data Field. Paired with each command is a data field. In the data field enter either a signed double integer or the number of a parameter (0-255), as appropriate for the configured command. Parameters 1-20 can be loaded with data using the LOAD-P command. Block number parameters can be entered in the data field when paired with Block or Jump instructions. The data field of a dwell command indicates the number of milliseconds that the program will pause in execution.

Axis# Field. This field is used to set the axis number for commands such as VELOC or PMOVE.

Table 4-4. Program Zero Motion Program Commands

Command Name	Definition	Logicmaster Range	Logicmaster Default
(NULL)	Do nothing	0	0
BLOCK	Define Block Number	1...65,535	1
BLOCK-SYNC	Define Synchronization Block Number	1...65,535	1
CMOVE-AL	Continuous move, Absolute, Linear	-8,388,608... 8,388,607	1
CMOVE-AL-P	Continuous move, Absolute, Linear, Use data in Parameter	1...255	1
CMOVE-AS	Continuous move, Absolute, S-curve	-8,388,608... 8,388,607	1
CMOVE-AS-P	Continuous move, Absolute, S-curve, Use data in Parameter	1...255	1
CMOVE-IL	Continuous move, Incremental, Linear	-8,388,608... 8,388,607	1
CMOVE-IL-P	Continuous move, Incremental, Linear, Use data in Parameter	1...255	1
CMOVE-IS	Continuous move, Incremental, S-curve	-8,388,608... 8,388,607	1
CMOVE-IS-P	Continuous move, Incremental, S-curve, Use data in Parameter	1...255	1
PMOVE-AL	Positioning move, Absolute, Linear	-8,388,608... 8,388,607	1
PMOVE-AL-P	Positioning move, Absolute, Linear, Use data in Parameter	1...255	1
PMOVE-AS	Positioning move, Absolute, S-curve	-8,388,608... 8,388,607	1
PMOVE-AS-P	Positioning move, Absolute, S-curve, Use data in Parameter	1...255	1
PMOVE-IL	Positioning move, Incremental, Linear	-8,388,608... 8,388,607	1
PMOVE-IL-P	Positioning move, Incremental, Linear, Use data in Parameter	1...255	1
PMOVE-IS	Positioning move, Incremental, S-curve	-8,388,608... 8,388,607	1
PMOVE-IS-P	Positioning move, Incremental, S-curve, Use data in Parameter	1...255	1
VELOC	Set Velocity	1...8,388,607	2000
VELOC-P	Set Velocity to data in Parameter	1...255	1
ACCEL	Set Acceleration	1...134,217,727	5000
ACCEL-P	Set Acceleration to data in Parameter	1...255	1
WAIT	Wait for CTL XX bit to go high before moving	1...12	1
LOAD-P01 ↓ **LOAD-P20**	Load DSM302 Parameter register number	-8,388,608... 8,388,607	0
DWELL	Wait for specified time in milliseconds	0...65,535	0
DWELL-P	Wait X milliseconds, X is the value in the Parameter	1...255	1
JUMP-UNCOND	Continue program execution at given block #	1...65,535	1
JUMP-CTL01 ↓ **JUMP-CTL12**	When the given CTL bit goes high during the current block execution, transfer program execution to the given block #	1...65,535	1

Program Zero Motion Command Descriptions

Each DSM302 command is briefly described below. For a comprehensive explanation of Motion Programming on the DSM302, refer to Chapters 7, 8, and 9 of this manual. See also, GFK-0664, *The Series 90 APM Programmer's Manual*.

Acceleration (ACCEL). This modal command is used to specify the axis acceleration and deceleration rate for subsequent moves. Once encountered, the specified rate will remain in effect until overridden by a later *Acceleration* command.

Block Number (BLOCK). Block numbers are used to monitor and synchronize program execution, terminate jump testing, and identify jump destinations. Block number is an actual command in the Program Zero Editor. The default block number is zero. Block numbers serve as labels for MOVE, DWELL, and WAIT commands.

Cmove (CMOVE). Continuous Move - this command is used when it is not necessary for the axis to be within the configured *In Position Zone* before proceeding to the next command. If no previous acceleration or velocity has been specified in a motion program, the configured *Jog Acceleration* and/or *Jog Velocity* will be used.

Dwell (DWELL). This command causes motion to cease for a specified time period (in milliseconds) before proceeding to the next command.

Jump (JUMP). This command is used to Branch program execution to another location in the program under certain specified states of the Faceplate control inputs (CTL01-CTL08) and %Q control outputs (CTL09-CTL12). The jump will occur when the condition tests "true" (logic 1). An unconditional *Jump* can also be selected. The jump may be forward or backward in the program. The jump condition will be tested as soon as the move prior to the *Jump* command has completed. A maximum of 220 jumps can exist for all programs and subroutines.

Once the condition testing is allowed to start, the test will occur once every two milliseconds until a Block Number or another Jump command is encountered. This will allow continued testing while a move or series of moves takes place, if the *Jump* command is located ahead of the moves in the block. If a jump occurs during a move, the remainder of the move is aborted and the command at the destination location is immediately effective. Jump destinations must be limited to the bounds of the program containing the *Jump* command.

Load Parameter (LOAD). This command initializes or changes a DSM302 data parameter value. The new value becomes effective immediately when encountered in the program. Program Zero may load parameters 1 to 20 only.

Pmove (PMOVE). Positioning Move - this command is used when it is necessary for the axis to be within the configured *In Position Zone* before proceeding to the next command. If no previous acceleration or velocity has been specified in a motion program, the configured *Jog Acceleration* and/or *Jog Velocity* will be used. The axis movement will stop for a minimum of two milliseconds in the *In Zone* range, then the next program command will execute.

Velocity (VELOC). This modal command specifies the velocity of axis motion. Once encountered, this command will remain in effect until overridden by a later *Velocity* command.

Wait (WAIT). This command synchronizes the start of axis motion with an external input or event reported in CTL01-CTL12. The start of motion is suspended until the bit being monitored is true.

Chapter 5 Motion Mate DSM302 to PLC Interface

This chapter defines the data that is transferred between the CPU and the Motion Mate DSM302 automatically each PLC sweep, without user programming. This data is categorized as follows:

- Input Status Data (Transferred from Motion Mate DSM302 to CPU)
  - Status Bits: 64 bits of %I data
  - Status Words: 40 words of %AI data, typically for standard mode applications
    50 words of %AI data, typically for follower mode applications
    64 words of %AI data, typically for applications requiring additional analog inputs
- Output Command Data (Transferred from CPU to Motion Mate DSM302)
  - Discrete Commands: 64 bits of %Q data
  - Immediate Commands: 6 words of %AQ data, typically for standard mode applications
    9 words of %AQ data, typically for follower mode applications
    12 words of %AQ data, typically for applications requiring additional analog outputs

Note

Throughout this chapter words shown in *italics* refer to actual PLC machine data references (%I, %A, %AI, %AQ).

Section 1: %I Status Bits

The following %I Status Bits are transferred automatically from the DSM302 to the CPU each sweep. The actual addresses of the Status Bits depend on the starting address configured for the %I references (see Table 4-1, "Module Configuration Data"). The bit offsets listed in the following table are offsets to this starting address. All reference section designations pertain to this chapter.

Table 5-1. %I Status Bits

Bit Offset	Description	Axis	Ref.	Bit Offset	Description	Axis	Ref.
00	Module Error Present	N/A	1.01	32	Axis Enabled	Servo 2	1.05
01	PLC Control Active	N/A	1.02	33	Position Valid	Servo 2	1.06
02	New Configuration. Received	N/A	1.03	34	Drive Enabled	Servo 2	1.07
03	*Reserved*			35	Program Active	Servo 2	1.08
04	CTL01, (+) Over travel (24v)	Servo 1	1.04	36	Moving	Servo 2	1.09
05	CTL02, (-) Over travel (24v)	Servo 1	1.04	37	In Zone	Servo 2	1.10
06	CTL03, Home Switch (24v)	Servo 1	1.04	38	Position Strobe 1 (5v)	Servo 2	1.11
07	CTL04, Strobe 1 Level (5v)	Servo 1	1.04	39	Position Strobe 2 (5v)	Servo 2	1.11
08	CTL05, (+) Over travel (24v)	Servo 2	1.04	40	Position Error Limit	Servo 2	1.12
09	CTL06, (-) Over travel (24v)	Servo 2	1.04	41	Torque Limit	Servo 2	1.13
10	CTL07, Home Switch (24v)	Servo 2	1.04	42	Servo Ready / IN4_B (5v)	Servo 2	1.14
11	CTL08, Strobe 1 Level (5v)	Servo 2	1.04	43	*Reserved*		
12	CTL13 / IN9_C (24v)	Aux 3	1.04	44	Follower Enabled	Servo 2	1.15
13	CTL14 / IN10_C (24v)	Aux 3	1.04	45	Follower Velocity Limit	Servo 2	1.16
14	CTL15 / Home Switch (24v)	Aux 3	1.04	46	*Reserved*		
15	CTL16 / Strobe 1 Level (5v)	Aux 3	1.04	47	*Reserved*		
16	Axis Enabled	Servo 1	1.05	48	Axis Enabled	Aux 3	1.05
17	Position Valid	Servo 1	1.06	49	Position Valid	Aux 3	1.06
18	Drive Enabled	Servo 1	1.07	50	*Reserved*		
19	Program Active	Servo 1	1.08	51	*Reserved*		
20	Moving	Servo 1	1.09	52	*Reserved*		
21	In Zone	Servo 1	1.10	53	*Reserved*		
22	Position Strobe 1 (5v)	Servo 1	1.11	54	Position Strobe 1 (5v)	Aux 3	1.11
23	Position Strobe 2 (5v)	Servo 1	1.11	55	Position Strobe 2 (5v)	Aux 3	1.11
24	Position Error Limit	Servo 1	1.12	56	*Reserved*		
25	Torque Limit	Servo 1	1.13	57	*Reserved*		
26	Servo Ready / IN4_A (5v)	Servo 1	1.14	58	IN4_C Input	Aux 3	1.14
27	*Reserved*			59	*Reserved*		
28	Follower Enabled	Servo 1	1.15	60	Input IN9_D (24v)	Aux 4	1.04
29	Follower Velocity Limit	Servo 1	1.16	61	Input IN10_D (24v)	Aux 4	1.04
30	*Reserved*			62	Input IN11_D (24v)	Aux 4	1.04
31	*Reserved*			63	Input IO5_D (5v)	Aux 4	1.04

1.01 Module Error Present. This status bit is set when the DSM302 detects any error. Errors related to Servo Axis 1 will be identified in the *Servo Axis 1 Error Code* %AI word. Errors related to Servo Axis 2 will be identified in the *Servo Axis 2 Error Code* %AI word. In follower mode, errors related to the Aux Axis 3 position input will be identified in the *Aux Axis 3 Error Code* %AI word. Module errors not related to a specific axis will be identified in the *Module Status Code* %AI word. See section 2, "%AI Status Words", for more details. The *Clear Error* %Q bit is the only command that will clear the *Module Error Present* %I status bit and the associated *Module Status Code* and *Servo Axis n Error Code* %AI word(s). If the condition causing the error is still present, the *Module Error Present* %I status bit will not be cleared.

1.02 PLC Control Active. Normally the *PLC Control Active* status bit is set, indicating that the %Q discrete commands or %AQ immediate commands from the PLC can control the DSM302. PLC Control Active is cleared only when the Status screen in the Motion Programmer is used instead of the PLC to control the DSM302, a capability not yet implemented.

1.03 New Configuration Received. The *New Configuration Received* %I status bit is set whenever the PLC sends a reset command or new configuration to the DSM302. *New Configuration Received* should be cleared by a PLC program before any %AQ Immediate commands such as *In Position Zone* or *Position Loop Time Constant* have been sent to the DSM302. The status bit can then be monitored by the PLC. If the bit is set, then the DSM302 has been reset or reconfigured. The PLC should clear the bit and then re-send all necessary %AQ commands. The bit is cleared by %AQ Immediate command 49h. Refer to section 4, "%AQ Immediate Commands," later in this chapter, for more details about the %AQ immediate command interfaces.

1.04 Faceplate Input %I Status Bits. These inputs indicate the state of external input devices connected to faceplate signals. The CTL01-CTL08 inputs (as well as CTL09-CTL12 outputs from the PLC %Q table) may be tested during the execution of *Wait* and *Conditional Jump* commands. CTL01-CTL08 and CTL13-CTL16 are all selectable as the Follower Enable Trigger bit in Follower mode. The Aux Axis 4 Inputs are reported to the PLC but are not available for Wait / Jump Control or Follower Enable Trigger.

Table 5-2. Faceplate Input %I Status Bits

Bit Name	Signal Name	Signal Use	Input Type	Faceplate Connector Pin	Digital Servo TB Pin	Wait / Jump Control	Follower Enable Trigger
CTL01	*IN9_A*	Servo Axis 1 (+) Overtravel	24v	A-16	6	Yes	Yes
CTL02	*IN10_A*	Servo Axis 1 (-) Overtravel	24v	A-34	14	Yes	Yes
CTL03	*IN11_A*	Servo Axis 1 Home Switch	24v	A-17	7	Yes	Yes
CTL04	*IN1_A*	Servo Axis 1 Strobe 1 Level	5v	A-1,19	1,9	Yes	Yes
CTL05	*IN9_B*	Servo Axis 2 (+) Overtravel	24v	B-16	6	Yes	Yes
CTL06	*IN10_B*	Servo Axis 2 (-) Overtravel	24v	B-34	14	Yes	Yes
CTL07	*IN11_B*	Servo Axis 2 Home Switch	24v	B-17	7	Yes	Yes
CTL08	*IN1_B*	Servo Axis 2 Strobe 1 Level	5v	B-1,19	1,9	Yes	Yes
CTL13	*IN9_C*	Aux Axis 3 Spare 24v input	24v	C-16	6	No	Yes
CTL14	*IN10_C*	Aux Axis 3 Spare 24v input	24v	C-34	14	No	Yes
CTL15	*IN11_C*	Aux Axis 3 Home Switch	24v	C-17	7	No	Yes
CTL16	*IN5_C*	Aux Axis 3 Strobe 1 Level	5v	C-9	N/C*	No	Yes
IN9_D	*IN9_D*	Aux Axis 4 Spare 24v input	24v	D-16	6	No	No
IN10_D	*IN10_D*	Aux Axis 4 Spare 24v input	24v	D-34	14	No	No
IN11_D	*IN11_D*	Aux Axis 4 Spare 24v input	24v	D-17	7	No	No
IN5_D	*IN5_D*	Aux Axis 4 Spare 5v input	5v	D-9	N/C*	No	No

* *If the Digital Servo Terminal Board IC693ACC335 is used for DSM Connector C or D, the IN5 signal is not available. If the Auxiliary Terminal Board IC693ACC336 is used, the terminal numbers are the same as the DSM faceplate connector pin numbers.*

1.05 Axis Enabled. The *Axis Enabled* status bit is ON when the DSM302 is ready to receive commands and control a servo. An error condition that stops the servo will turn *Axis Enabled* OFF. **When *Axis Enabled* is OFF, no commands other than the *Clear Error* %Q bit will be accepted by the axis.**

1.06 Position Valid. The *Position Valid* status bit indicates that a *Set Position* command or successful completion of a Find Home cycle has initialized the home reference value in the *Actual Position* % AI status word. *Position Valid* must be ON in order to execute a motion program.

If the DSM302 is configured to use an absolute feedback digital encoder (GE Fanuc α or β Series servo with optional encoder battery), *Position Valid* is automatically set whenever the digital encoder reports a valid absolute position. See Appendix C for details of operation when absolute mode digital encoders are used.

1.07 Drive Enabled. The *Drive Enabled* status bit indicates the state of the *Enable Drive* %Q bit and the solid state relay output supplied by the DSM302. The ON state of the *Drive Enabled* %I bit corresponds to the CLOSED state of the relay output and the ON state of the associated faceplate EN LED. In Digital mode, the solid state relay provides the MCON signal to the GE Fanuc Digital Servo through the servo command cable. *Drive Enabled* is cleared following power-up or an error condition that stops the servo.

1.08 Program Active. The *Program Active* status bit for each axis indicates that a Motion Program (0-10) or a *Move* %AQ command (27h) is executing on that axis. Executing a multi-axis program will set both *Program Active* %I bits.

1.09 Moving. The *Moving* status bit is ON when *Commanded Velocity* is non-zero; otherwise, it is OFF (i.e. the *Moving* bit will transition to OFF while an axis is steadily following a Master input). All *Move* (positioning and continuous), *Jog*, Accel and Decel Ramps, and *Move at Velocity* commands will cause the *Moving* bit to be set to ON. The *Force Digital Servo Velocity* %AQ command will not set the *Moving* bit.

In FOLLOWER mode, *Moving* is ON for the conditions described above and is not affected by the enabled or disabled state of the Master Axis input. *Moving* is also ON whenever the Follower Acceleration Ramp feature is active. Refer to Chapter 8, Follower Motion, for additional information on the Follower Acceleration Ramp.

1.10 In Zone. *In Zone* can be used to determine when the servo has arrived at its destination. Operation of the *In Zone* bit depends on the *Position Error* value and the state of the *Moving* %I bit:

Table 5-3. In Zone Bit Operation

Cmd Generator Active (*Moving* %I bit ON)	*Position Error* ≤ In Position Zone	*In Zone* bit
No	Yes	ON
No	No	OFF
Yes	No	OFF
Yes	Yes	OFF

In Zone will be ON whenever *Moving* is OFF and *Position Error* is less than or equal to the configured ***In Position Zone*** value. This condition occurs at the end of each *Positioning Move* (PMOVE) command or any time the axis commands are halted and *Actual Position* has approached *Commanded Position* (e.g. for *Dwell*, *Feed Hold*, or *Feedrate* Override % = 0).

In Follower mode, *In Zone* is ON under the same conditions described above. If the *Follower Enabled* %I bit is ON and the slave axis is following the master axis input with no superimposed motion, *In Zone* is ON if *Position Error* is ≤ *In Zone* value (i.e. the following axis is closely following the master). If *Moving* is ON due to a superimposed *Jog*, *Move at Velocity, Accel or Decel Ramp,* or *Execute Program* command, *In Zone* will always be off.

1.11 Position Strobe 1, Position Strobe 2. The *Position Strobe 1* and *Position Strobe 2* status bits indicate that an OFF to ON transition has occurred at the associated faceplate Strobe Input. When this occurs, an axis position is captured and reported in the *Strobe n Position* %AI status word, where "n" is Axis 1, Axis 2, or Axis 3. The *Position Strobe* %I bit is cleared by the associated *Reset Strobe n* %Q bit. **A maximum of 2 PLC sweeps is required for the *Position Strobe* %I bit to be cleared in the PLC after a *Reset Strobe n* %Q bit is turned ON.** Once the *Position Strobe n* bit is cleared, new data may be captured by another Strobe Input. Position data will be captured within 250 microseconds from a Strobe Input.

Note

The *Position Strobe* bits do not indicate the logic level of the faceplate input, they only indicate that an OFF to ON transition has occurred on the input.

1.12 Position Error Limit. The *Position Error Limit* status bit is set when the absolute value of the position error exceeds the configured *Position Error Limit* value. When the *Position Error Limit* status bit is set, commanded velocity and commanded position are frozen to allow the axis to ”catch up” to the commanded position.

1.13 Torque Limit. The *Torque Limit* status bit is set when the commanded torque exceeds the torque limit setting for the configured motor type.

1.14 Servo Ready (Servo Axis 1,2), IN4_C (Aux Axis 3). This status bit is set when faceplate signal IN4 of the associated connector (A, B or C) is ON (active low: ON = 0v, OFF = +5v). For Servo Axis 1 and Servo Axis 2, this input reports the *Servo Ready* state of the servo amplifier. For Aux Axis 3, The IN4_C input is a spare input to the PLC.

1.15 Follower Enabled. This status bit indicates when the Follower is enabled for the axis. The *Enable Follower* % Q bit and an optional CTL01-CTL16 faceplate trigger input enable the Follower function. If follower ramp acceleration control is active when *Follower Enabled* turns on, the axis will accelerate to the master velocity command, and when it turns off, the axis will decelerate to zero master velocity command. Both acceleration and deceleration during the ramp process will utilize the currently active Jog acceleration.

1.16 Follower Velocity Limit. The *Follower Velocity Limit* status bit is set if the velocity requested by the external or internal master input exceeds the configured *Velocity Limits*. Therefore, *Follower Velocity Limit* is an indication that the follower axis is no longer locked to its master input. The error is reported in the associated *Servo Axis n Error Code* %AI word when *Follower Velocity Limit* is set.

An exception exists when one of the Velocity Limits is set to 0. A zero Velocity Limit indicates unidirectional following motion so no error is generated for the limit that is set to zero. For example, if the + Velocity Limit = 0 and + Counts are input, the *Follower Velocity Limit* Status bit is set, but no Status Error code is reported.

If one of the Follower Control Loop Velocity Limits is set to zero, the position error integrator is inhibited from generating motion in the direction of the Velocity Limit. This feature allows unidirectional systems to operate properly with the position error integrator enabled.

Section 2: %AI Status Words

The following %AI Status Words are transferred automatically from the DSM302 to the CPU each sweep. The total number of the %AI Status Words is configured with the Configuration Software to be a length of 40, 50, or 64. The actual addresses of the Status Words depend on the starting address configured for the %AI references. See Table 4-1, Module Configuration Data. The word numbers listed in the following table are offsets to this starting address. All reference section designations pertain to this chapter. All %AI data except *Actual Velocity* is updated within the DSM302 once every 2 milliseconds. *Actual Velocity* is updated once every 128 milliseconds.

Table 5-4. %AI Status Words

Word Offset	Description	Axis	Ref	Word Offset	Description	Axis	Ref
					(Configured for 50 %AI words)		
00	Module Status Code	N/A	2.01	40-41	Actual Position.	Aux 3	2.05
01	Servo Axis 1 Error Code	Servo 1	2.02	42-43	Strobe 1 Position	Aux 3	2.06
02	Servo Axis 2 Error Code	Servo 2	2.02	44-45	Strobe 2 Position	Aux 3	2.06
03	Aux Axis 3 Error Code	Aux 3	2.02	46-47	Actual Velocity	Aux 3	2.09
04	*Reserved*			48-49	*Reserved*		
					(Configured for 64 %AI words)		
05	Command Block Number	Servo 1	2.03	50	Connector A Analog Input 1	Servo 1	2.12
06-07	Commanded Position	Servo 1	2.04	51	Connector A Analog Input 2	Servo 1	2.12
08-09	Actual Position	Servo 1	2.05	52	Connector B Analog Input 1	Servo 2	2.12
10-11	Strobe 1 Position	Servo 1	2.06	53	Connector B Analog Input 2	Servo 2	2.12
12-13	Strobe 2 Position	Servo 1	2.06	54	Connector C Analog Input 1	Aux 3	2.12
14-15	Position Error	Servo 1	2.07	55	Connector C Analog Input 2	Aux 3	2.12
16-17	Commanded Velocity	Servo 1	2.08	56	Connector D Analog Input 1	Aux 4	2.12
18-19	Actual Velocity	Servo 1	2.09	57	Connector D Analog Input 2	Aux 4	2.12
20-21	User Selected Data	Servo 1	2.10	58-63	*Reserved*		
22	User Selected Analog Input	N/A	2.11				
23	Command Block Number	Servo 2	2.03				
24-25	Commanded Position	Servo 2	2.04				
26-27	Actual Position	Servo 2	2.05				
28-29	Strobe 1 Position	Servo 2	2.06				
30-31	Strobe 2 Position	Servo 2	2.06				
32-33	Position Error	Servo 2	2.07				
34-35	Commanded Velocity	Servo 2	2.08				
36-37	Actual Velocity	Servo 2	2.09				
38-39	User Selected Data	Servo 2	2.10				

2.01 Module Status Code. *Module Status Code* indicates the current operating status of the DSM302. When the *Module Error Present* %I flag is set, and the error is not related to a specific axis, an error code number is reported in the *Module Status Code* that describes the condition causing the error.

For a list of Motion Mate DSM302 error codes refer to Appendix A.

2.02 Servo Axis 1, Servo Axis 2, Aux Axis 3 Error Code. The *Servo Axis 1 Error Code*, *Servo Axis 2 Error Code* and *Aux Axis 3 Error Code* words indicate the current operating status of Servo Axis 1, Servo Axis 2 and Aux Axis 3 (Follower Master), respectively. When the *Module Error Present* %I flag is set, and the error is related to a particular axis, an error code number is reported which describes the condition causing the error.

For a list of Motion Mate DSM302 error codes refer to Appendix A.

2.03 Command Block Number. *Command Block Number* indicates the block number of the command that is presently being executed in the active Program or Subroutine. It changes at the start of each new block as the program commands are executed, and thus identifies the present operating location within the program. Block numbers are displayed only if the motion program uses them. Additionally, the most recently used block number will be displayed until superseded by a new value.

2.04 Commanded Position. *Commanded Position* (user units) is where the axis is commanded to be at any instant in time. The difference between *Commanded Position* and *Actual Position* is the *Position Error* value that produces the Velocity Command to drive the axis. The rate at which the *Commanded Position* is changed determines the velocity of axis motion.

If *Commanded Position* moves past either of the count limits, it will roll over to the other limit and continue in the direction of the axis motion.

2.05 Actual Position. *Actual Position* (user units) is a value maintained by the DSM302 to represent the physical position of the axis. It is set to an initial value by the *Set Position* %AQ Immediate command or to *Home Position* by the *Find Home* cycle. When digital absolute encoders are used, *Actual Position* is automatically set whenever the encoder reports a valid position. The motion of the axis feedback device continuously updates the axis *Actual Position*.

If *Actual Position* moves past either of the count limits, it will *roll over* to the other limit and continue in the direction of the axis motion.

2.06 Strobe 1, 2 Position. *Strobe 1 Position* and *Strobe 2 Position* (user units) contain the axis actual position when a Strobe 1 Input or Strobe 2 Input occurs. When a Strobe Input occurs, the *Position Strobe 1* or *Position Strobe 2* %I bit is set to indicate to the PLC that new Strobe data is available in the related *Strobe 1 Position* or *Strobe 2 Position* status word. The PLC must set the proper *Reset Strobe 1* or *Reset Strobe 2* Flag %Q bit to clear the associated *Position Strobe n* %I bit.

Strobe 1, 2 Position will be maintained and will not be overwritten by additional Strobe Inputs until the *Position Strobe 1, 2* %I bit has been cleared. If the *Reset Strobe Flag* %Q bit is left in the ON state (thus holding *Position Strobe 1, 2* input flags in the cleared state), then each Strobe Input that occurs will cause the axis position to be captured in *Strobe 1, 2 Position*.

The *Strobe 1, 2 Position* actual position values are also placed in data parameter registers for use with motion programs commands. The data parameter register assignments are as follows:

	Servo Axis 1	Servo Axis 2	Aux Axis 3
Strobe 1 Position	P224	P232	P240
Strobe 2 Position	P225	P233	P241

This feature allows the strobe input to trigger a Conditional JUMP in a program block using the *Strobe 1 Position* or *Strobe 2 Position* as the destination of a CMOVE or PMOVE command.

See Chapter 1, "Product Overview, DSM302 Position Strobes," for information on strobe latency and processing times.

2.07 Position Error. *Position Error* (user units) is the difference between *Commanded Position* and *Actual Position*. In the servo control loop, *Position Error* is multiplied by a gain constant to provide the servo velocity command.

2.08 Commanded Velocity. *Commanded Velocity* (user units/sec) is a value generated by the DSM302 axis command generator. *Commanded Velocity* indicates the instantaneous velocity command that is producing axis motion. At the beginning of a move it will increase at the acceleration rate, and once the programmed velocity has been reached, it will stabilize at the programmed velocity value.

In Follower mode, *Commanded Velocity* only represents the output of the axis command generator. The Follower Master Axis input or the Follower Acceleration Ramp controller does not affect *Commanded Velocity*.

2.09 Actual Velocity. *Actual Velocity* (user units/sec) represents the axis velocity derived from the Feedback device and is updated by the DSM302 once every 250 milliseconds.

2.10 User Selected Data. The information reported in *User Selected Data* is determined by the *Select Return Data* %AQ command. Refer to Section 4 “%AQ Immediate Commands” for additional information. *User Selected Data* has these defaults:

Servo Axis 1 Data: Servo Axis 1 Commanded Torque (always 0 in Analog mode)

Servo Axis 2 Data: Servo Axis 2 Commanded Torque (always 0 in Analog mode)

2.11 User Selected Analog Input. This %AI word reports a single analog input value that could be selected from one of the eight analog inputs available in the DSM302. The input data is scaled so that +/- 32000 = +/- 10.00v. The default input is Analog Input 1 of Aux Axis 3. The ability to select any other analog input is not currently available.

2.12 Connector “A” - “D” Analog Inputs 1, 2. When the %AI Length in the module configuration is set to 64 words, the values of all eight analog inputs are reported in word offsets 50-57 of the %AI table. The data is scaled so that +/- 32000 = +/- 10.00v.

When Servo Axis 1, 2 (Connectors A and B) control a digital servo, the analog inputs are connected through the servo command cable to the servo amplifier. These inputs detect the AC servo phase feedback currents and are not available for general use. If a Servo Axis is not used to control a digital servo, an Auxiliary Terminal Board IC693ACC336 and associated cable may be used to allow analog input connections to user devices.

For Aux Axis 3,4 (Connectors C and D) the analog inputs are available for connection to user devices. An Auxiliary terminal board IC693ACC336 and associated cable must be used for this purpose.

On each faceplate connector and Auxiliary Terminal Board, Analog Input 1 connects to pins 7 (+) and 25 (-). Analog Input 2 connects to pins 8 (+) and 26 (-).

Section 3: %Q Discrete Commands

The following %Q Outputs represent Discrete Commands that are sent automatically to the DSM302 from the CPU each PLC sweep. A command is executed simply by turning on the Output Bit of the desired command. The actual addresses of the Discrete Command bits depend on the starting address configured for the %Q references. See Table 4-1, Module Configuration Data. The Bit Offsets listed in the following table are offsets to this starting address. All reference section designations pertain to this chapter.

Table 5-5. %Q Discrete Commands

Bit Offset	Description	Axis	Ref	Bit Offset	Description	Axis	Ref
00	Clear Error	N/A	3.01	32	Abort All Moves	Servo 2	3.04
01	Execute Program 0	N/A	3.02	33	Feed Hold (Pause Program)	Servo 2	3.05
02	Execute Program 1	N/A	3.02	34	Enable Drive / MCON)	Servo 2	3.06
03	Execute Program 2	N/A	3.02	35	Find Home	Servo 2	3.07
04	Execute Program 3	N/A	3.02	36	Jog Plus	Servo 2	3.09
05	Execute Program 4	N/A	3.02	37	Jog Minus	Servo 2	3.10
06	Execute Program 5	N/A	3.02	38	Reset Strobe 1	Servo 2	3.11
07	Execute Program 6	N/A	3.02	39	Reset Strobe 2	Servo 2	3.11
08	Execute Program 7	N/A	3.02	40	OUT1_B Output Control	Servo 2	3.12
09	Execute Program 8	N/A	3.02	41	OUT3_B Output Control	Servo 2	3.13
10	Execute Program 9	N/A	3.02	42	*Reserved*		
11	Execute Program 10	N/A	3.02	43	*Reserved*		
12	CTL09 Program Control	N/A	3.03	44	Enable Follower	Servo 2	3.14
13	CTL10 Program Control	N/A	3.03	45	Select Follower Int Master	Servo 2	3.15
14	CTL11 Program Control	N/A	3.03	46	*Reserved*		
15	CTL12 Program Control	N/A	3.03	47	*Reserved*		
16	Abort All Moves	Servo 1	3.04	48	*Reserved*		
17	Feed Hold (Pause Prgm)	Servo 1	3.05	49	*Reserved*		
18	Enable Drive / MCON	Servo 1	3.06	50	*Reserved*		
19	Find Home	Servo 1	3.07	51	Find Home	Aux 3	3.08
20	Jog Plus	Servo 1	3.09	52	*Reserved*		
21	Jog Minus	Servo 1	3.10	53	*Reserved*		
22	Reset Strobe 1	Servo 1	3.11	54	Reset Strobe 1	Aux 3	3.11
23	Reset Strobe 2	Servo 1	3.11	55	Reset Strobe 2	Aux 3	3.11
24	OUT1_A Output Control	Servo 1	3.12	56	OUT1_C Output Control	Aux 3	3.12
25	OUT3_A Output Control	Servo 1	3.13	57	OUT3_C Output Control	Aux 3	3.13
26	*Reserved*			58	*Reserved*		
27	*Reserved*			59	*Reserved*		
28	Enable Follower	Servo 1	3.14	60	OUT1_D Output Control	Aux 4	3.12
29	Select Follower Int Master	Servo 1	3.15	61	OUT3_D Output Control	Aux 4	3.13
30	*Reserved*			62	*Reserved*		
31	*Reserved*			63	*Reserved*		

3.01 Clear Error. When an error condition is reported, this command is used to clear the *Module Error Present* %I status bit and the associated *Module Status Code*, Servo *Axis 1 Error Code*, Servo *Axis 2 Error Code and Aux Axis 3 Error Code* %AI status words. Error conditions that are still present (such as an *End of Travel* limit switch error) will not be cleared and must be cleared by some other corrective action. If the *Clear Error* bit is maintained ON, a Jog command can be used to move away from an open hardware overtravel limit switch.

3.02 Execute Motion Program 0 - 10. These commands are used to select stored programs for immediate execution. Each command uses a one shot action; thus a command bit must transition from OFF to ON each time a program is to be executed. Programs may be temporarily interrupted by a *Feed Hold* command.

When a program begins execution, Rate Override is always set to 100%**.** A *Rate Override* %AQ command can be sent on the same sweep as the *Execute Motion Program n* %Q bit and will be effective as the program starts.

Only one Motion Program can be executed at a time per axis. The *Program Active* %I status bit must be OFF or Motion Program execution will not be allowed to start. A multi-axis Motion Program uses both axis 1 and axis 2, so both *Program Active* bits must be OFF to start a multi-axis Motion Program.

3.03 CTL09-CTL12 Control Bits. These command bits may be tested by the DSM302 during execution of *Wait* or *Conditional Jump* commands.

3.04 Abort All Moves. This command causes <u>any</u> motion in progress to halt at the current *Jog Acceleration* rate. Any pending programmed or immediate command is canceled and therefore not allowed to become effective. The abort condition is in effect as long as this command is on. If motion was in progress when the command was received, the *Moving* status bit will remain set and the *In Zone* status bit will remain cleared until the commanded velocity reaches zero and the *In Zone* condition is achieved.

3.05 Feed Hold (On Transition). This command causes any motion programs in progress to stop at the active acceleration rate. (This Feed Hold command does not stop motion commanded by a master source in Follower Mode.) Once the motion is stopped, the *Moving* status bit is cleared and the *In Zone* status bit is set when the I*n Zone* condition is attained. *Jog* commands are allowed when in the Feed hold condition. After an ON transition, program motion will stop, even if the command bit transitions back OFF before motion stops.

Feed Hold (Off Transition). This command causes any motion programs interrupted by *Feed Hold* to resume at the programmed acceleration and velocity rate. Additional program moves will then be processed and normal program execution will continue. *Feed Hold* OFF behaves in a similar fashion to an Execute Program command except the path generation software uses only the remaining distance in the program.

If jogging occurred while *Feed Hold* was ON, the interrupted *Move* command will resume from where the axis was left after the *Jog*. The *Move* finishes at the correct programmed velocity and continues to the original programmed position as if no *jog* displacement occurred.

3.06 Enable Drive / MCON. If the *Module Error Present* and *Drive Enabled* %I status bits are cleared, this command will cause the Drive Enable relay contact to close and the *Drive Enabled* bit to be set. When the *Drive Enabled* %I bit is set, the path generation and position control functions are enabled and servo motion can be commanded. A signal will be sent (*MCON*) to the digital servo enabling the drive. *Enable Drive* must be maintained ON to allow normal servo motion (except when using *Jog* commands).

3.07 Find Home (Servo Axis 1, 2). This command causes the DSM302 to establish the ***Home Position***. A Home Limit Switch Input from the I/O connector roughly indicates the reference position for Home and the next encoder marker encountered indicates the exact home position. When the ***Home Mode*** axis configuration is set to MOVE+ or MOVE-, the Home Limit Switch input will be ignored. The configured ***Home Offset*** defines the location of ***Home Position*** as the offset distance from the Home Marker. The *Position Valid* %I bit indication is set at the conclusion of the Home Cycle. See Chapter 6 for more information regarding the Home Cycle. See Appendix C for considerations when using an absolute encoder.

3.08 Find Home (Aux Axis 3). This command does not cause any motion; instead it initializes the Aux Axis 3 Encoder logic to wait for a marker pulse transition when the Aux Axis 3 Home Switch input is ON. When the Home Switch / Marker condition is satisfied, the Aux Axis 3 *Actual Position* will be set to the configured ***Home Position*** and the Aux Axis 3 *Position Valid* %I bit will be set. The Home Switch input may be disabled by the ***Home Mode*** axis configuration, allowing the Home Cycle to only wait for a marker transition. *Find Home* must be maintained ON until the marker is located. Clearing the *Find Home* %Q bit before the marker is located will end the Home Cycle without setting Aux Axis 3 *Position Valid* %I bit.

3.09 Jog Plus. When this command bit is ON, the axis moves in the positive direction at the configured *Jog Acceleration* and *Jog Velocity* rates. The axis will move as long as the *Jog Plus* command is maintained and the configured ***Positive End Of Travel*** software limit or Positive Overtravel switch is not encountered. The Overtravel switch inputs can be disabled using the ***OT Limit*** configuration parameter. ***Jog Plus* may be used to jog off of the Negative Overtravel switch if the *Clear Error* %Q bit is also maintained on.** <u>See Chapter 6, Non-Programmed Motion, for more information on Jogging with the DSM302.</u>

3.10 Jog Minus. When this command bit is ON, the axis moves in the negative direction at the configured *Jog Acceleration* and *Jog Velocity* rates. The axis will move as long as the *Jog Minus* command is maintained and the configured ***Negative End Of Travel*** software limit or Negative Overtravel switch is not encountered. The Overtravel switch inputs can be disabled using the ***OT Limit*** configuration parameter. ***Jog Minus* may be used to jog off of the Positive Overtravel switch if the *Clear Error* %Q bit is also maintained on.** <u>See Chapter 6, Non-Programmed Motion, for more information on Jogging with the DSM302.</u>

3.11 Reset Strobe 1, 2 Flag. The *Position Strobe n* %I status bit flag informs the PLC that a Strobe Input has captured an axis position that is now stored in the associated *Strobe n Position* %AI status word. When the PLC acknowledges this data, it may use the *Reset Strobe n* Flag %Q command bit to clear the *Position Strobe n* %I status bit flag. Once the *Position Strobe n %I* bit is set, additional Strobe Inputs will not cause new data to be captured. The flag must be cleared before another Strobe Position will be captured. As long as the *Reset Strobe n* Flag %Q command bit is set, the *Position Strobe n* bit flag will be held in the cleared state. In this condition, the latest Strobe Input position is reflected in the *Strobe n Position* status word, although the flag cannot be used by the PLC to indicate when new data is present.

3.12 OUT1_A, B, C, D Output Control. Each axis connector has a 24-vdc solid state relay (SSR) output rated at 125 ma. *The OUT1A, OUT1_B, OUT1_C* and *OUT1_D Output Control* %Q bits control the state of the associated output. For each axis the following connector terminals are assigned:

	Faceplate Connector Pin	Auxiliary TB IC693ACC336 Terminal	Servo TB IC693ACC335 Terminal
OUT1 SSR (+) terminal	18	18	8
OUT1 SSR (-) terminal	36	36	16

3.13 OUT3_A, B, C, D Output Control. Each axis connector has a differential 5-vdc output that is suitable for driving 5v TTL or CMOS loads. The *OUT3_A, OUT3_B, OUT3_C* and *OUT3_D Output Control* %Q bits control the state of the associated output. For each axis the following connector pins are assigned:

	Faceplate Connector Pin	Auxiliary TB IC693ACC336 Terminal	Servo TB IC693ACC335 Terminal
OUT3 (+) terminal	14	14	5
OUT3 (-) terminal	32	32	13

3.14 Enable Follower. When this bit is set and the *Follower Enabled* %I status bit indicates the Follower is enabled, motion commanded by the external or internal master will act as an input to the follower loop. An optional Follower Trigger bit may be configured to initiate follower motion. When a Follower Trigger is used, *Enable Follower* must be ON for the trigger condition to be tested. Clearing *Enable Follower* disconnects the follower loop from the master source. *Jog, Move at Velocity,* and *Execute Program n* commands will be allowed regardless of the state of *Enable Follower*. When the Follower is enabled, *Jog*, *Move at Velocity,* or *Execute Program n* commands will be superimposed on the master velocity or position command. *Find Home* is not allowed unless *Enable Follower* is cleared. Refer to Chapter 8 for additional information. This bit is only used by follower mode.

3.15 Select Follower Internal Master. This bit switches the master axis source from Encoder 3 to the Internal Master Velocity generator. The *Internal Master Velocity* %AQ command can be used to change the velocity of the internal master. This bit is only used by follower mode.

Section 4: %AQ Immediate Commands

The following %AQ Immediate Command words are transferred each PLC sweep from the CPU %AQ data to the DSM302. The length of the %AQ Immediate Command words is configured with the Configuration Software to be a length of 6, 9, or 12. The actual addresses of the Immediate Command words depend on the starting address configured for the %AQ references. See Table 4-1, Module Configuration Data. The word numbers listed in the following table are offsets to this starting address. The words are assigned as follows:

Table 5-6. %AQ Word Assignments

Word Offset	Description	Axis
00	Immediate Command Word	Servo 1
01-02	Command Data	Servo 1
03	Immediate Command Word	Servo 2
04-05	Command Data	Servo 2
	Configured for 9 %AQ words	
06	Immediate Command Word*	Aux3
07-08	Command Data*	Aux3
	Configured for 12 %AQ words	
09	Immediate Command Word*	Aux4
10-11	Command Data*	Aux4

* These words can be used for *Force Analog Output* commands ONLY

Only one %AQ Immediate command may be sent to each axis of the DSM302 every PLC sweep. The only exception is the *Load Parameter Immediate* command, which is axis independent. Thus two *Load Parameter Immediate* commands can be sent on the same sweep (one in the first three %AQ words and the other in the second three %AQ words).

Even though the commands are sent each sweep, the DSM302 will act on a command ONLY if it changed since the last sweep. When any of the 3 words change, the DSM302 will accept the data as a new command and respond accordingly.

The *Axis Enabled* %I bit must be ON for an axis to accept a new %AQ Immediate Command. Under some conditions such as a disconnected encoder or un-powered servo amplifier, *Axis Enabled* will be OFF and the %AQ command processing for that axis will be disabled. If Servo Axis 1 or 2 is not used for motor control, the configured ***Motor Type*** must be set to 0 or an error will be reported and *Axis Enabled* will stay OFF.

The 6-byte format for the Immediate Commands is defined in Table 5-7. The actual addresses of the Immediate Command Words depend on the starting address configured for the %AQ references. **The word numbers listed in the following table are offsets to this starting address.**

The word offsets are shown in reverse order and in hexadecimal to simplify the data entry. The following example sends the Set Position command to axis 1. The first word, word 0, contains the actual command number. For the Set Position command, the command number is 0023h. The

second and third words contain the data for the Set Position command that is a position. The second word, word 1, is the least significant word of the position and the third word, word 2, is the most significant word.

Example:

To set a position of 3,400,250, first convert the value to hexadecimal. 3,400,250 decimal equals 0033E23A hexadecimal. For this value, 0033 is the most significant word and E23A is the least significant word. The data to be sent to the DSM302 would be:

Word 2	***Word 1***	***Word 0***	***Command***
0033	E23A	0023	Set Position 3,400,250

Setting up word 0 as a hexadecimal word and words 1 and 2 as a double integer in a LM90 Mixed Reference Table Display will simplify immediate command entry.

In the following %AQ command table, only the word offsets for Servo Axis 1 are listed. Word offsets for the other axes are computed by adding 3 (Servo Axis 2), 6 (Aux Axis 3), or 9 (Aux Axis 4) to the listed word offsets. Note that the only Immediate Command that may be used with Aux Axis 3 or 4 is *Force Analog Output*. The Ref column numbers refer to sections in this chapter.

Table 5-7. %AQ Immediate Commands Using the 6-Byte Format

Word 2		Word 1		Word 0			
Byte 5	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Byte 0**	**Immediate Command Definition**	**Ref**
xx	xx	xx	xx	00	00h	**Null**	**4.01**
xx	xx	xx	RO%	00	20h	**Rate Override** RO% = 0 ...120%	**4.02**
xx	xx	*	Incr	00	21h	**Position Increment Without Position Update** Incr. = -128 ... +127 User Units	**4.03**
Velocity				00	22h	**Move At Velocity** Vel. = -8,388,608 ... + 8,388,607 User Units/sec	**4.04**
Position				00	23h	**Set Position** Pos. = -8,388,608 ... + 8,388,607 User Units	**4.05**
xx	xx	Analog Output		00	24h	**Force Analog Output** Analog Output = -32,000 ... + 32,000	**4.06**
xx	xx	*	Incr.	00	25h	**Position Increment With Position Update** Incr. = -128 ... +127 User Units	**4.07**
xx	xx	xx	In Posn Zone	00	26h	**In Position Zone** Range = 0 ... 255	**4.08**
Position or Parameter #				Move Type	27h	**Move Command** Pos. = -8,388,608 ... +8,388,607 User Units Par # = 0 ... 255	**4.09**
Velocity				00	28h	**Jog Velocity** Vel. = +1 ... + 8,388,607 User Units/sec	**4.10**
Acceleration				00	29h	**Jog Acceleration** Acc. = +1 ... + 134,217,727 User Units/sec/sec	**4.11**
xx	xx	Time Constant		00	2Ah	**Position Loop Time Constant** Time Constant = 0,5 ... 10000	**4.12**
xx	xx	xx	VFF%	00	2Bh	**Velocity Feedforward** VFF% = 0 ... 100%	**4.13**
xx	xx	Integr. TC		00	2Ch	**Integrator Time Constant** Time Constant = 0, 10 ... 10,000 ms	**4.14**
Ratio B		Ratio A		00	2Dh	**Follower A/B Ratio** Ratio A = –32,768 … +32,767 Ratio B = +1 ... +32,767	**4.15**
xx	xx	xx	VLGN	00	2Eh	**Velocity Loop Gain (Digital mode only)** VLGN = 0 ... 255	**4.16**
xx	xx	xx	Torque Limit %	00	2Fh	**Torque Limit (Digital mode only)** Range = 0-100%	**4.17**
Velocity				00	30h	**Internal Master Velocity** Vel = –1,000,000 ... + 1,000,000 counts/sec	**4.18**

Table 5-7. - Continued - %AQ Immediate Commands Using the 6-Byte Format

Word 2		Word 1		Word 0			
Byte 5	**Byte 4**	**Byte 3**	**Byte 2**	**Byte 1**	**Byte 0**	**Immediate Command Definition**	**Ref**
Position				00	31h	**Set Aux Axis 3 Position** Pos = –8,388,608 ... +8,388,607 counts	**4.19**
xx	xx	Digital Servo Velocity Cmd		00	34h	**Force Digital Servo Velocity** Servo Velocity Cmd = -4,095 ... +4,095 RPM	**4.20**
xx	xx	Offset		Mode	40h	**Select Return Data**	**4.21**
xx	xx	Make-Up Time		00	42h	**Follower Ramp Distance Make-Up Time** Active Range = 0, 10 ... 32000 ms	**4.22**
Position				00	43h	**Set Follower Winder Position** Pos = 0 ... 8,388,606 counts	**4.23**
Position				00	44h	**Set Follower Winder Zone Length** Pos = 100... 8,388,607 counts	**4.24**
xx	xx	Mode		Axis	47h	**Select Analog Output Mode (Digital mode only)**	**4.25**
xx	xx	xx	xx	00	49h	**Clear New Configuration Received**	**4.26**
Parameter Data				Par #h	50h	**Load Parameter Immediate** Par # = 0 ... 255 Parameter Data = Range depends on parameter usage.	**4.27**

* = Only 00 or FFh are acceptable.
xx = don't care

4.01 Null. This is the default %AQ Immediate command. Since the %AQ words are automatically transferred each PLC sweep, the *Null* command should always be used to avoid inadvertent execution of another %AQ Immediate command.

4.02 Rate Override. This command immediately changes the % feedrate override value, which will modify the commanded velocity for all subsequent programmed moves. This new value will become effective immediately when received by the DSM302. It is stored and will remain effective until overwritten by a different value. A rate override has no effect on non-programmed motion or acceleration. *Rate Override* is set to 100% whenever a program is initiated. The *Rate Override* command can be sent on the same PLC sweep as an *Execute Program* %Q bit and the Override value will immediately take effect. Rate Override can be used to adjust the programmed velocity (not acceleration) of a particular move or a set of moves on any given axis.

4.03 Position Increment Without Position Update. (User units) This command offsets the axis position from -128 to +127 user units without updating the *Actual Position* or *Commanded Position*. The DSM302 will immediately move the axis by the increment commanded if the servo is enabled. Position Increments can be used to make minor machine position corrections to compensate for changing actual conditions. See Chapter 6, Non-Programmed Motion, for more information on Position Increment Commands with the DSM302.

4.04 Move At Velocity. (User units/sec) This command is executed from the PLC to move the axis at a constant velocity. The active ***Jog Acceleration*** rate is used for *Move at Velocity* commands. Axis actual position data will roll over at the configured Hi or Lo Limit when reached during these moves. See Chapter 6, Non-Programmed Motion, for more information on the Move at Velocity Command.

4.05 Set Position. (User units) This command changes the axis position register values without moving the axis. The *Commanded Position* and *Actual Position* values will both be changed so that no motion command will be generated. The *Actual Position* will be set to the value designated and the *Commanded Position* will be set to the value + *Position Error*. *Set Position* cannot be performed when the *Moving* %I bit or the *Program Active* %I bit is ON. In Follower mode, *Set Position* is not allowed if the *In Zone* %I bit is OFF. In Standard mode, *Set Position* is allowed if the *In Zone* %I bit is OFF as long as *Actual Velocity* is ≤ 100 cts/sec. The position value must be within the End of Travel Limits and Count Limits or a status error will be reported. The *Position Valid* %I bit is set after a successful *Set Position* command. See Appendix C for considerations when using absolute mode encoders. The Set Position command is commonly used to set the starting position reference point to zero (or another value) without homing the axis.

Note

When a GE Fanuc digital servo system is first powered up after removal or replacement of the encoder battery, the digital encoder must be rotated past its internal reference point. If this is not done the *Set Position* command will be ignored and Error Code 53h (Set Position before encoder passes reference point) will be reported.

4.06 Force Analog Output. The *Force Analog Output* immediate command operates one of the analog outputs on DSM faceplate connector C or D in Digital mode, or in Analog mode, on connector A, B, C, or D. Multiple *Force Analog Output* commands can be used to operate outputs on different connectors by using the appropriate %AQ word offsets (see the paragraph before Table 5-7). A *Force Analog Output* value of +32000 will produce +10.00 Vdc and a *Force Analog Output* value of -32000 will produce -10.00 Vdc. Note that in Digital mode, a *Force Analog Output* signal can be overridden if another signal is routed to the same connector by the *Select Analog Output Mode* command. However, the *Select Analog Output Mode* command can also be used to restore an overridden *Force Analog Output* signal. See Section 4.25, "Select Analog Output Mode."

Force Analog Output must remain continuously in the %AQ data for proper operation. When a Force Analog Output command is active for a given axis, any other %AQ immediate command for that axis will remove the Force Analog command and turn off the associated analog output.

4.07 Position Increment With Position Update. (User units) This command is similar to the *Position Increment Without Position Update* command (#21h) except that *Actual Position* and *Commanded Position* (returned in %AI data) are both updated by the increment value. If the servo is enabled, the DSM302 will immediately move the axis by the increment value. Position Increments can be used to make minor machine position corrections to compensate for changing actual conditions. See Chapter 6, Non-Programmed Motion, for more information on Position Increment Commands with the DSM302.

4.08 In Position Zone. This command can be used to set the active ***In Position Zone*** to a value different than the configured value.

The DSM302 compares ***In Position Zone*** to the *Position Error* in order to control the *In Zone* %I bit. When the *Moving* %I bit is OFF and *Position Error* is ≤ ***In Position Zone***, the *In Zone* %I bit is ON.

If the DSM302 is power cycled or the PLC CPU is reset for any reason, the value set by this command will be lost and the *In Position* zone value set by configuration software will be reinstated.

4.09 Move Command. This command will produce a single move profile that will move the axis to the position commanded each time it is sent. The current ***Jog Acceleration*** and ***Jog Velocity*** (which can also be changed by %AQ commands) will be used for the move.

The data field for this command may contain the move position or distance in bytes 2-5 with the command type (in hexadecimal format) as defined below:

Move Type (byte 1):

00h = Abs, Pmove, Linear

01h = Abs, Cmove, Linear

10h = Abs, Pmove, Scurve

11h = Abs, Cmove, Scurve

40h = Inc, Pmove, Linear

41h = Inc, Cmove, Linear

50h = Inc, Pmove, Scurve

51h = Inc, Cmove, Scurve

The data field for this command may contain a parameter number in byte 2 (bytes 3-5 unused) with the command type as defined below:

Move Type (byte 1):

80h = Abs, Pmove, Linear

81h = Abs, Cmove, Linear

90h = Abs, Pmove, Scurve

91h = Abs, Cmove, Scurve

C0h = Inc, Pmove, Linear

C1h = Inc, Cmove, Linear

D0h = Inc, Pmove, Scurve

D1h = Inc, Cmove, Scurve

The *Move* *Command* is executed as a single move motion program. Therefore all the restrictions that apply to motion program execution also apply to the *Move Command*. For example, if a program is already active for axis 1, then an attempt to send this command for axis 1 will result in an error condition being reported.

4.10 Jog Velocity. (User units/sec) This command sets the velocity used when a *Jog* %Q bit is used to jog in the positive or negative direction. A PLC reset or power cycle returns this value to the configured data. Additionally, the Jog Velocity is used as the default velocity for motion programs.

4.11 Jog Acceleration. (User units/sec/sec) This command sets the acceleration value used by *Jog*, *Move at Velocity*, a *Find Home Cycle*, *Move at Velocity, Follower Ramp Control,* and *Abort All Moves*. Additionally, the Jog Acceleration is used as the default acceleration for motion programs. A PLC reset or power cycle returns this value to the configured data.

Note: A minimum value after scaling is used in the DSM302. This value is determined by the rule: Jog Acc * (user units/counts) >= 32 counts/sec/sec.

4.12 Position Loop Time Constant. (Milliseconds) This command allows the servo position loop time constant to be changed from the configured value. The lower the Position Loop Time Constant value, the faster the system response. Values that are too low will cause system instability and oscillation. For accurate tracking of the commanded velocity profile, the ***Position Loop Time Constant*** should be 1/4 to 1/2 of the MINIMUM system acceleration or deceleration time. The ***Vel at 10 V*** configuration value must be set correctly for proper operation of the ***Position Loop Time Constant***. A PLC reset or power cycle returns this value to the configured data.

4.13 Velocity Feedforward. This command sets the ***Velocity Feedforward*** gain (percent). It is the percentage of *Commanded Velocity* that is added to the DSM302 velocity command output. Increasing ***Velocity Feedforward*** causes the servo to operate with faster response and reduced position error. Optimum ***Velocity Feedforward*** values are 90-100 %. The "Vel at 10 V" configuration value must be set correctly for proper operation of the ***Velocity Feedforward*** gain factor. A PLC reset or power cycle returns this value to the configured data.

4.14 Integrator Time Constant. (Milliseconds) This command sets the ***Integrator Time Constant*** for the position error integrator. The value specifies the amount of time in which 63% of the *Position Error* will be removed. The ***Integrator Time Constant*** should be 5 to 10 times greater than the ***Position Loop Time Constant*** to prevent instability and oscillation. **It is recommended that the position error integrator only be used in continuous follower applications. Use of the integrator in point to point positioning applications may result in position overshoot when stopping.**

4.15 Follower A/B Ratio. This command allows the PLC to update the slave **:** master A/B ratio used in each follower loop. "A" is a 16-bit signed integer with a minimum value of -32,768 and a maximum value of +32,767. "B" is a 16-bit integer with a minimum value of 1 and a maximum value of 32,767. The magnitude of the A/B ratio must be in the range 32:1 to 1:10,000 or a status error will be generated. (Applies to firmware version 1.2 or later. For earlier firmware versions, the ratio must be in the range of 32:1 to 1:32). Refer to Chapter 8 for additional information about the A/B ratio.

4.16 Velocity Loop Gain. (VLGN) Digital Mode only. The velocity control loop gain for a GE Fanuc digital servo axis may be set with the *Velocity Loop Gain* command. The VLGN value is used to match the load inertia (J_L) to the motor inertia (J_M). VLGN is defined with a default value of 16 representing an inertia ratio of 1 to 1. The VLGN value is calculated assuming that the load is rigidly applied to the motor. Therefore, in actual machine adjustment the required value may significantly differ from the calculated value due to rigidity, friction, backlash, and other factors. A PLC reset or power cycle returns VLGN to the value set in the configuration software. A suggested starting point for *Velocity Loop Gain* is:

$$\text{Velocity Loop Gain} = \frac{\text{Load Inertia } (J_L) * 16}{\text{Motor Inertia } (J_M)}$$

The allowed range of *Velocity Loop Gain* is 0 to 255.

For example: The motor inertia (J_M) of a particular servo is 0.10 lb-in-s^2. The load inertia (J_L) in this application is 0.05 lb-in-s^2. VLGN = (0.05 / 0.10) $*$ 16 = 8

The default *Velocity Loop Gain* is set using the ***Vel Lp Gain*** setting in the Logicmaster Configuration Software.

Caution

An incorrect VLGN value may cause an axis to be unstable. Care should be used when making any change to the VLGN value.

4.17 Torque Limit (TRLMT). Digital Mode only. The *Torque Limit* Command provides a method of limiting the torque produced by the GE Fanuc servomotor. The DSM302 will set the torque limit (TRLMT) at the default 100% whenever a power cycle or reset occurs. The PLC application logic must set any other value for desired TRLMT. The valid range for TRLMT is 0 to 100 % (of peak torque at commanded velocity) in units of 1%. TRLMT can be changed during axis motion and takes effect immediately. Refer to the appropriate servo motor manual for the motor torque curve to determine the actual value of torque output available at a given velocity. A simple example would be the use of Torque Limit to prevent over-tightening on a machine.

4.18 Internal Master Velocity. (Counts/sec) This command loads the FOLLOWER mode Internal Master velocity generator that may be used as an alternate source for the follower Master input. The ***Mstr Source*** axis configuration must be set to ENC3/INT, and the **Select Follower Int Master** %Q bit must be set ON, to make use of the Internal Master velocity generator. The maximum allowed velocity is +/- 1,000,000 counts/sec. There is only one Internal Master velocity generator for use by all axes.

4.19 Set Aux Axis 3 Position. (Counts) This command sets the actual position value for the encoder on Aux Axis 3 without using a Find Home operation. The *Position Valid* %I bit for Aux Axis 3 will be set when the command is received.

4.20 Force Digital Servo Velocity. (RPM) Digital Mode only. This command bypasses the position loop and forces a velocity command to the digital servo for tuning purposes. Acceleration control is not used and changes in velocity take effect immediately. A *Force Digital Servo Velocity* command value of +4095 will produce a motor velocity of + 4,095 RPM and -4095 will produce a motor velocity of -4,095 RPM (depending on individual motor maximum velocities). The digital servo control loops may limit actual motor speed to a lower value. Care should be taken not to operate a servomotor past the rated duty cycle.

The *Enable Drive* %Q bit must be active with no other motion commanded for the *Force Digital Servo Velocity* command to operate. The command must remain continuously in the %AQ data for proper operation. When a Force Digital Servo Velocity command is active for a given axis, any other %AQ immediate command for that axis will remove the *Force Digital Servo Velocity* data and halt the servo. Chapter 6, Non-Programmed Motion, also contains information on Force Digital Servo Velocity.

4.21 Select Return Data. This command allows alternate data to be reported in the *User Selected Data* %AI location for each axis. The alternate data includes information such as Parameter memory contents and the DSM302 Firmware Revision.

The *Select Return Data* command uses a mode selection and an offset selection. The mode selection (byte offset +1 of the six byte command) determines the Return Data type. The offset selection (byte offsets +2, +3 of six-byte command) selects an individual data item for some modes. Setting the mode to 00h causes the default Torque Command to be reported. The following Return Data selections are allowed:

Digital	Analog	Selected Return Data	Data Mode	Data Offset
Y	N	Torque Command	00h	not used
Y	Y	DSM Firmware Revision	10h	not used
Y	Y	DSM Firmware Build ID No. (hex)	11h	not used
Y	N	Absolute Feedback Offset (cts)	17h	not used
Y	Y	Parameter Data	18h	Parameter Number (0–255)
Y	Y	Analog Inputs - Axis 1	1Ch	not used
Y	Y	Analog Inputs - Axis 2	1Dh	not used
Y	Y	Analog Inputs - Aux 3	1Eh	not used
Y	Y	Analog Inputs - Aux 4	1Fh	not used
Y	Y	Commanded Position (user units)	20h	not used
Y	Y	Follower Program Command Position (cts)	21h	not used
Y	Y	Follower Winder Zone Position (cts)	22h	not used

Torque Command is scaled so that +/- 7282 = +/- 100% torque.

DSM Firmware Revision is interpreted as two separate words for major-minor revision codes.

DSM Firmware Build ID is interpreted as a single hex word.

Absolute Feedback Offset is the position offset (in counts) that is used to initialize *Actual Position* when a GE Fanuc digital Absolute Encoder is used. *Actual Position* = Absolute Encoder Data + Absolute Feedback Offset.

Analog Inputs provides two words of data for each axis: low word = AIN1 and high word = AIN2. The data is scaled so that +/- 32000 = +/- 10.0v.

Refer to Chapter 8, "Follower Motion", for an explanation of *Follower Program Command Position* and *Follower Winder Zone Position*.

At least three PLC sweeps or 20 milliseconds (whichever represents more time) must elapse before the new Selected Return Data is available in the PLC.

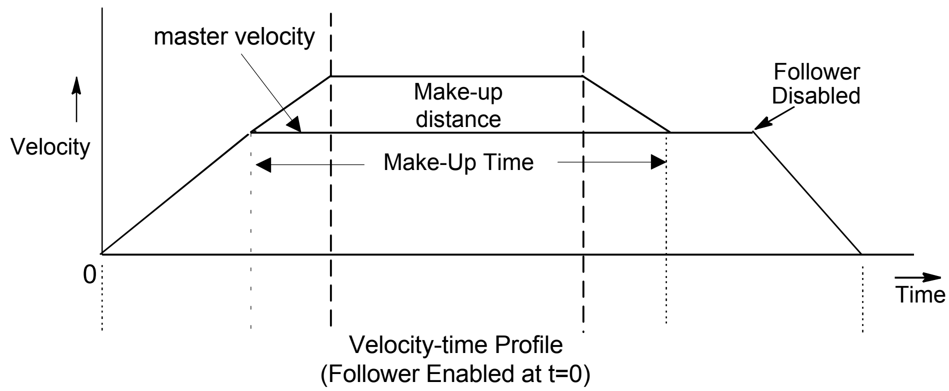
4.22 Follower Ramp Distance Make-Up Time. When the Follower Ramp feature has been selected (FOLLOWR MODE is configured for ACC RAMP) and the follower is enabled, the following axis is ramped up to the Master velocity at the current ***Jog Acceleration*** rate when the Master Velocity is non-zero at the time the Follower is enabled. The master counts that accumulate during acceleration of the follower axis are stored. In this mode, the follower axis will accelerate to a velocity that exceeds the Master Velocity in order to make up the position error that accumulated while the Follower axis was accelerating to the Master Velocity. This make-up distance correction has a trapezoidal velocity profile determined by the ***Follower Ramp Distance Make-Up Time*** and jog acceleration at the beginning of the correction. This mode is used when the Follower axis must be position-and-velocity-synchronized to the Master position at the instant the Follower mode was enabled.

If the ***Follower Ramp Distance Make-Up Time*** is too short then the velocity profile is a triangular profile. If during the distance correction, velocity exceeds 80% of the velocity limit, then the automatically calculated velocity will be clamped at 80% of the configured velocity limit. In both

cases a warning message is reported and the real distance make-up time is longer than programmed, but the distance is still corrected properly.

Setting a ***Follower Ramp Distance Make-Up Time*** of 0 allows the Ramp feature to accelerate the axis without making up any of the accumulated counts. In this instance, the Follower axis velocity will not exceed the master velocity. For applications where the Follower axis only needs to be synchronized to the master velocity and lost counts do not matter, set the distance make-up time = 0.

Typical velocity profile during the follower ramp cycle is shown below.



See Chapter 8, “Follower Motion, Follower Axis Acceleration Ramp Control” section, for a much more detailed discussion of this feature.

4.23 Set Follower Winder Position. (Counts) This command sets the Winder Position when a follower axis is operating in WINDER mode. The position must be within the present *Winder Zone Length* boundaries or an error will be reported. Refer to Chapter 8, “Follower Motion”, for a detailed explanation of the WINDER mode.

4.24 Set Follower Winder Zone Length. (Counts) This command sets the Winder Zone length when the follower axis is operating in WINDER mode. An error will be reported if a previous Zone Length change has not yet been processed. An error will also be reported if the follower is enabled and the Zone Length change is greater than 25 %. Refer to Chapter 8, “Follower Motion”, for a detailed explanation of the WINDER mode.

4.25 Select Analog Output Mode. **Digital Mode only.**

For GE Fanuc digital servos, this command lets you choose what analog signals will be sent to the Analog Output pins (pins 6 and 24) on the four DSM faceplate connectors. The *Select Analog Output Mode* command uses a Signal Code to specify the signal to be sent, and a Connector Code to specify the DSM connector to receive the signal. This command is particularly useful for servo tuning.

Use the following structure to set up your 6-byte %AQ Immediate Command (described in Table 5-7):

- Byte 0 contains the Select Analog Output Mode command code (47h).
- Byte 1 contains the Connector Code, a hex number.
- Byte 3 contains the Signal Code, a decimal number.
- Bytes 4, 5, and 6 are not used and should contain Zeroes.

Connector Codes

Connector Code	Connector Selected	Connector Pins
01h	Connector A	Pin 6 = OUT Pin 24 = COM (Ref. to 0V) Refer to the I/O Connection Diagrams in Chapter 3 for Terminal Board connections.
02h	Connector B	
03h	Connector C	
04h	Connector D	

Signal Codes

The following table lists the Signal Codes. Note that some signals have a default output, while others do not. The *Select Analog Output Mode* command lets you (1) re-route two of the signals that have a default output (the %AQ *Force Analog Output* data signal cannot be re-routed), (2) gain access to signals lacking a default output, and (3) restore signals to their default outputs.

Signal Code	Signal Description	Default Output to:
00 decimal*	%AQ *Force Analog Output* data*	Connector C or D
10 decimal	Servo Axis 1 Torque Command	None
15 decimal	Servo Axis 1 Actual Velocity	Connector A
20 decimal	Servo Axis 2 Torque Command	None
25 decimal	Servo Axis 2 Actual Velocity	Connector B

* Cannot be re-routed. This signal code can only be used to restore this signal back to its default output.

Example:

To select Servo Axis 1 Torque Command as the Analog Output on Connector D, place the following data in the %AQ immediate command words:

Word 2 (Byte 5 and Byte 4)	Word 1 (Byte 3 and Byte 2)	Word 0 (Byte 1 and Byte 0)
+00000	+00010	0447h

Signal Code (10)

Target Connector Code (04h)

Select Analog Output Mode Command Code (47h)

4.26 Clear New Configuration Received. This command clears the *New Configuration Received* %I bit. Once cleared, the *Configuration Complete* bit is only set when the PLC resets or reconfigures the module. The PLC can monitor the bit to determine if it must re-send other %AQ commands, such as *In Position Zone* or *Jog Acceleration*. This would only be necessary if the %AQ commands were used to override DSM302 configuration data programmed with the PLC configuration software.

4.27 Load Parameter Immediate. This command is executed from the PLC to immediately change a DSM302 data parameter value. It can be sent from either Axis #1 or Axis #2 Command registers. Data parameters are only used by motion programs. A command for each parameter change is required. Note that Parameter 0 is not supported by all Configuration and Programming Software packages.

Chapter 6 Non-Programmed Motion

The DSM302 can generate motion in an axis in one of six ways without the use of any motion programs.

- *Find Home* and *Jog Plus/Minus* use the %Q bits to command motion.
- *Move at Velocity*, *Move*, *Force Digital Servo Velocity*, *Force Analog Output*, and *Position Increment* use %AQ immediate commands.

During *Jog*, *Find Home, Move at Velocity, Move* and *Force Digital Servo Velocity*, any other commanded motion, programmed or non-programmed, will generate an error. The only exception is the *Position Increment* %AQ command, which can be commanded any time. See the description of *Position Increment* motion below for more details.

Non-programmed motions (*Abort All Moves*, *Jog Plus/Minus*, *Move at Velocity* and *Move*) use the ***Jog Acceleration*** and ***Jog Acceleration Mode***. The *Feed* Hold %Q command uses the programmed acceleration and acceleration mode.

DSM302 Home Cycle

A home cycle can be used to establish a correct Actual Position relative to a machine reference point. The configured **Home Offset** defines the location of *Home Position* as the offset distance from the Home Marker.

The *Enable Drive* %Q bit must be ON during an entire home cycle. However, the *Find Home* %Q bit does not need to be held ON during the cycle; it may be turned on momentarily with a one-shot. Note that turning ON the *Find Home* %Q bit immediately turns OFF the *Position Valid* %I bit until the end of the home cycle. The *Abort All Moves* %Q bit halts a home cycle, but the *Position Valid* bit does not turn back ON. No motion programs can be executed unless the *Position Valid* bit is ON.

Home Switch Mode

If the ***Find Home Mode*** is configured as HOMESW (HOME Switch), the Home Switch input from the axis I/O connector is used first to roughly indicate the reference position for home. Then, the next encoder marker encountered when traveling in the negative direction indicates the exact location. An open Home Switch input indicates the servo is on the positive side of the home switch and a closed Home Switch input indicates the axis is on the negative side of the home switch. An

OFF to ON transition of the *Find Home* %Q command yields the following home cycle. Unless otherwise specified, acceleration is at the current ***Jog Acceleration*** and configured ***Jog Acceleration Mode***.

Find Home Routine for Home Switch

If initiated from a position on the positive side of the home switch, in which case the home switch must be OPEN (Logic 0), the Find Home routine starts with step 1 below. (All of the first several steps of the following routine are necessary to allow for a variety of possible home switch designs and starting positions.) If the Find Home routine is initiated from a position on the negative side of the home switch, in which case the home switch must be CLOSED (Logic 1), the routine starts with step 3 below.

1. The axis is moved in the negative direction at the configured ***Find Home Velocity*** until the Home Switch input closes.
2. The axis decelerates and is stopped.
3. The axis is accelerated in the positive direction and moved at the configured ***Find Home Velocity*** until the Home Switch input opens.
4. The axis decelerates and is stopped.
5. The axis is accelerated in the negative direction and moved at the configured ***Final Home Velocity*** until the Home Switch input closes.
6. The axis continues negative motion at the configured ***Final Home Velocity*** until a marker pulse is sensed. The marker establishes the home reference position.
7. The axis decelerates and is stopped.
8. The axis is moved, at the current ***Jog Velocity***, the number of user units specified by the ***Home Offset*** value from the home reference position.
9. The axis decelerates and is stopped.
10. The DSM302 sets the *Commanded Position* and *Actual Position* %AI status words to the configured ***Home Position*** value. Finally, the DSM302 sets the *Position Valid* %I bit to indicate the home cycle is complete.

Home Switch Example

Many different home switch designs are possible. The switch may be normally open or normally closed, and may be mounted in one of several possible locations. The example given in this section illustrates a fairly common arrangement used for linear axes. In the following picture, the home switch is a normally open proximity switch, mounted near the end of the machine slide's travel range (in the negative direction). The imaginary line that divides the home switch's positive and negative sides is the home switch's operating point, located approximately on the switch's centerline. If the machine slide travels in the negative direction far enough so that the right-hand edge of the home switch cam causes the home switch to close, we consider the machine slide as having crossed over to the "negative side" of the home switch. The home switch cam is long enough so that while the machine slide is on the negative side of the home switch, it will keep the normally open home switch closed.

Note the relationships of the home position, the negative overtravel position, and the positive stop position. A small amount of distance is provided in the negative direction between the home position and the negative overtravel position. This is to allow some "working room" for adjustment

and setup of these positions and for the “find home” routine, which requires that its final move be in the negative direction.

Distance is also provided between the overtravel limit position and the positive stop. Enough distance should be allowed here to prevent the machine slide from hitting the positive stop. The correct distance needs to be greater than the worst-case stopping distance required by the machine slide after it reaches the overtravel limit position.

In this example, the machine slide’s working range is on the positive side of the home switch. If the DSM’s *Home Position* parameter was set to 0, this would simplify programming absolute positioning commands since only positive numbers would be used.

Often, the home position needs to be set to an exact distance from a reference point on the machine. To facilitate this adjustment, the home switch cam could be made with slotted mounting holes that would allow a coarse adjustment of the cam to bring the calibration to within one turn of the encoder. Then, the small remaining distance would be accurately measured and the value obtained would be entered into the DSM’s *Home Offset* parameter.

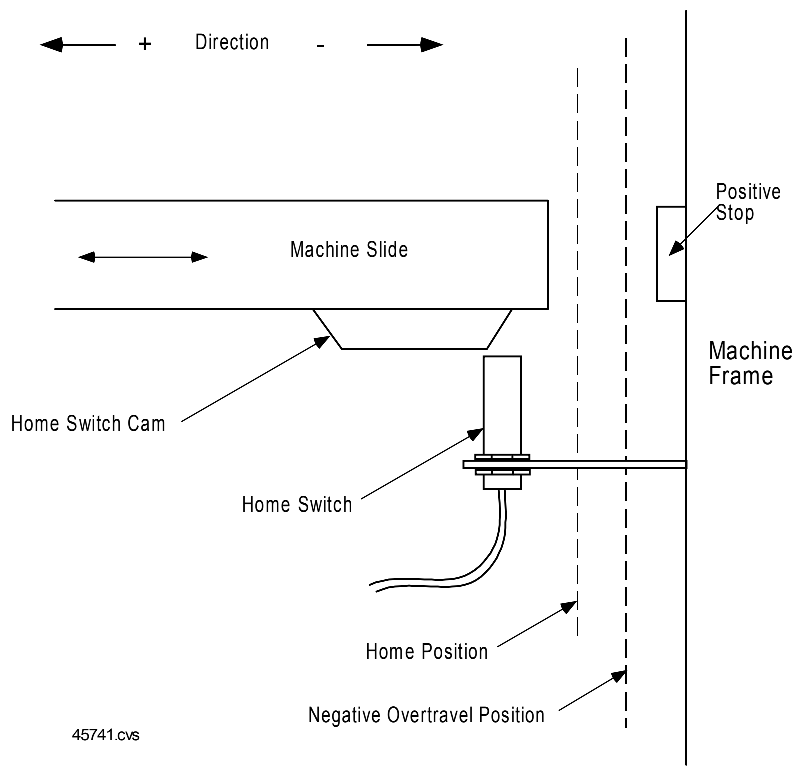


Figure 6-1. Home Switch Example

Move+ and Move– Modes

If ***Find Home Mode*** is configured as MOVE+ or MOVE–, the first encoder marker pulse encountered when moving in the appropriate direction (positive for MOVE+, negative for MOVE–) after the find home command is given is used to establish the exact location. In this mode, the operator usually jogs the axis to a position close (within one revolution of the encoder) to the home

position first, then initiates the find home command. To assist the operator in jogging to the correct position, a set of alignment marks indicating a close proximity to the home position is sometimes placed on the machine and machine axis. The next picture shows an example of the Home Position parameter set to Move – (minus). In this example, the operator jogs the axis until the moveable mark on the machine slide lines up with the stationary mark on the alignment plate mounted to the machine frame. (Note that the marks align on the positive side of home position since the Home Position parameter is set to Move –). Then the operator initiates the find home routine, which causes the axis to move in the negative direction until the marker pulse occurs.

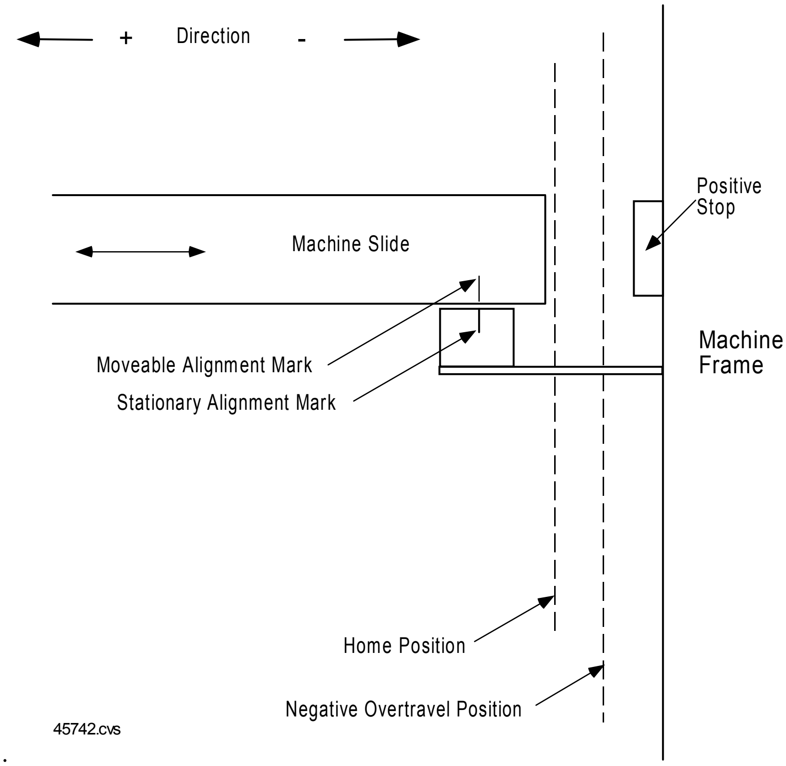


Figure 6-2. Move – (Minus) Home Position Example

Find Home Routine for Move + or Move –

When the find home command (an OFF to ON transition of the *Find Home* %Q bit) is initiated, the following sequence of events occurs:

1. The axis is accelerated at the ***Jog Acceleration*** *rate* and moved at the configured ***Final Home Velocity*** (positive direction for MOVE+, negative direction for MOVE–) until a marker pulse is sensed. This marker pulse establishes the home reference position.
2. The axis is stopped at the configured ***Jog Acceleration*** rate and with the configured ***Jog Acceleration Mode***.
3. The axis is moved, at the configured ***Jog Velocity*** and with the configured ***Jog Acceleration*** rate and ***Jog Acceleration Mode***, the number of user units specified by the ***Home Offset*** value from the home reference position.
4. The axis is stopped at the configured ***Jog Acceleration*** rate and with the configured ***Jog Acceleration Mode***.

5. The DSM302 sets the *Commanded Position* and *Actual Position* %AI status words to the configured ***Home Position*** value; the DSM302 sets the *Position Valid* %I bit to indicate the home cycle is complete.

Jogging with the DSM302

The ***Jog Velocity*, *Jog Acceleration***, and ***Jog Acceleration Mode*** are configuration parameters in the DSM302. These values are used whenever a *Jog Plus* or *Jog Minus* %Q bit is turned ON. Note that if both bits are ON simultaneously, no motion is generated. The ***Jog Acceleration*** and ***Jog Acceleration Mode*** are also used during a *Find Home* cycle and when a *Move at Velocity* %AQ immediate command is performed. Programmed motions use the ***Jog Velocity*** and ***Jog Acceleration*** as defaults.

A *Jog Plus/Minus* %Q command can be performed when no other motion is commanded, or while programmed motion is temporarily halted due to a *Feed Hold* %Q command. The *Enable Drive* %Q bit does not need to be ON to jog, but it can be ON. Turning on a *Jog Plus/Minus* %Q bit will automatically close the Enable Relay, and turn on the *Drive Enabled* %I bit. When an overtravel limit switch is OFF, *Jog Plus/Minus* and *Clear Error* %Q bits may be turned on simultaneously to move away from the open limit switch. Thus a *Jog Plus* %Q command will not work while the positive end of travel switch is open and *Jog Minus* will not work while the negative end of travel switch is open.

Move at Velocity Command

A *Move at Velocity* %AQ command is generated by placing the value 22h in the first word of %AQ data assigned to an axis. The second and third words together represent a signed 32-bit velocity. Note that the third word is the most significant word of the velocity. Once the command is given, the %AQ data can be cleared by sending a *NULL* command, or changed as desired. *Move at Velocit*y will not function unless the servo drive is enabled (*Enable Drive* %Q command and *Drive Enabled* %I status bit are set).

The listing of %AQ immediate commands shows the words in reverse order to make understanding easier. For example, to command a velocity of 512 user units per second in a DSM302 configured with %AQ data starting at %AQ1, the following values should be used: 0022h (34 decimal) in %AQ1, 0200h (512 decimal) in %AQ2, and 0 in %AQ3. When the DSM302 receives these values, if *Drive Enabled %I* is ON, *Abort All Moves %Q* is OFF, and no other motion is commanded it will begin moving the axis at 512 user units per second in the positive direction using the current ***Jog Acceleration*** and ***Acceleration Mode***.

The *Drive Enabled* %I bit must be ON before the DSM302 receives the immediate command or an error will occur. Also, if a *Move at Velocity* command is already in the %AQ data, the velocity value must change while the *Drive Enabled* bit is ON for the DSM302 to accept it. The DSM302 detects a *Move at Velocity* command when the %AQ values change.

When the DSM302 is performing a *Move at Velocity command*, it ignores the software end of travel limits (***Pos EOT*** and ***Neg EOT***). Hardware overtravel limits must be ON if they are enabled.

A *Move at Velocity* command can be stopped without causing an error in two ways: a *Move at Velocity* command with a velocity of zero, or turning the *Abort All Moves* %Q bit ON for at least one PLC sweep.

Force Digital Servo Velocity Command (DIGITAL Servos)

This command bypasses the position loop and forces a velocity RPM command to the *digital* servo for tuning purposes. Acceleration control is not used and changes in velocity take effect immediately. A *Force Digital Servo Velocity* command value of +4095 will produce a motor velocity of + 4,095 RPM and -4095 will produce a motor velocity of -4,095 RPM (depending on individual motor maximum velocities). The digital servo control loops may limit actual motor speed to a lower value.

CAUTION

Care should be taken not to operate a servomotor beyond its rated duty cycle.

The *Enable Drive* %Q bit must be active with no other motion commanded for the *Force Digital Servo Velocity* command to operate. The command must remain continuously in the %AQ data for proper operation. When a Force Digital Servo Velocity command is active for a given axis, any other %AQ immediate command for that axis will remove the *Force Digital Servo Velocity* data and halt the servo. A one-shot *Force Digital Servo Velocity* command will therefore only operate during the sweep in which it appears.

Refer to Chapter 5, Motion Mate DSM302 to PLC Interface, for more information on this command.

Note: The Force Analog Output command, described below, is used for analog servos.

Force Analog Output Command (ANALOG Servos)

In Analog mode, the *Force Analog Output* %AQ immediate command operates the analog output on the DSM faceplate connectors A, B, C, or D. A *Force Analog Output* value of +32000 will produce +10.00 Vdc and a *Force Analog Output* value of -32000 will produce -10.00 Vdc.

Force Analog Output operates only while the %AQ data is active. When a Force Analog Output command is active for a given axis, any other %AQ immediate command for that axis will remove the Force Analog Output command and turn off the associated analog output.

Refer to Chapter 5, “Motion Mate DSM302 to PLC Interface”, for more information on this command.

Position Increment Commands

To generate small corrections between the axis position and the DSM302 tracking, the *Position Increment* %AQ commands can be used to offset *Actual Position* by a specific number of user units. If the *Drive Enabled* %I bit is ON, the axis will immediately move the increment amount. If the position increment without position update is used (%AQ command 21h), the *Actual Position* %AI status word reported by the DSM302 will remain unchanged. If the *Position Increment With Position Update* is used (%AQ command 25h), the *Actual Position* and *Commanded Position* %AI status words reported by the DSM302 will be changed by the increment value. *Position Increment* can be used at any time, though simultaneous use with the *Force Digital Servo Velocity* command is impossible because the *Force Digital Servo Velocity* command must remain in the %AQ command data area or the servo will be stopped.

Other Considerations

Other considerations when using non-programmed motion are as follows:

- The *Abort All Moves* %Q bit, when ON, will prevent any non-programmed motion from starting.
- Turning ON the *Abort All Moves* %Q bit will immediately stop any current non-programmed motion at the current ***Jog Acceleration***.
- A *Set Position* %AQ command during non-programmed motion will cause a status error.
- Turning OFF the *Enable Drive* %Q bit while performing a home cycle or executing a *Move at Velocity* %AQ command will cause a stop error.
- The *Feed Hold* %Q bit has no effect on non-programmed motion.
- The *Rate Override* %AQ command has no effect on non-programmed motion.
- Changing the ***Jog Velocity*** or ***Jog Acceleration*** will not affect moves in progress.

Chapter 7 Programmed Motion

A motion program consists of a group of user-programmed motion command statements that are stored to and executed in the DSM302. The DSM302 executes motion program commands sequentially in a block-by-block fashion once a program is selected to run. The motion program is executed autonomously from the PLC, although the PLC starts the DSM302 motion program and can interface with it (with parameters and certain commands) during execution. In addition, external inputs (CTL bits) connected directly to the DSM302 faceplate can be used in motion programs to delay or alter program execution flow. The PLC receives status information (such as position, velocity, and Command Block Number) from the DSM302 during program execution. Program Zero is edited with the Logicmaster configuration software, and is incorporated into, and stored with the module configuration. Motion programs 1-10 and sub-routines 1-40 are edited with the Motion Programmer software and must be stored to the DSM302 via its COMM port connector, located on its faceplate. Please refer to the *Series 90 PLC APM Programmer's Manual*, GFK-0664, for further information.

Single-Axis Motion Programs and Sub-Routines

A single-axis program contains program statements for one axis only. The programmed axis is specified as a motion program command operand, either axis 1 or axis 2, in the motion program command statements. The DSM302 may operate two single-axis programs, one for each axis, independently or simultaneously. For example, motion program 1 may be written for axis one and motion program 2 written to control axis two. Each axis may be home referenced and the motion program for each axis may execute independently without regard to the state of the other axis. Alternately, program 1 and program 2 may execute simultaneously (via the run program %Q bits) during the same PLC sweep.

Motion programs 1-10 support the sub-routine feature, but Program Zero does not. Sub-routines may include all the available motion program commands including the *CALL* command. The *BLOCK-SYNC* command is reserved for multiaxis programs and sub-routines. Sub-routine "nesting" using *CALL* statements is supported to a maximum of 8 levels. Single axis sub-routines, similar to motion programs, contain commands for only one axis. The difference is that a single axis sub-routine does not specify the axis number as an a motion program operand command. A single axis motion program may *CALL* any single axis sub-routine stored in module memory. For example, single axis motion program 1, operating axis one, may include a CALL statement to single axis sub-routine 1. Additionally, single axis motion program 2, operating axis two, may include a CALL statement to single axis sub-routine 1. Single axis motion programs cannot *CALL* multiaxis sub-routines.

The motion program and sub-routine structure allow flexibility in execution and axis control in the DSM302 module. The practical limitation is that an axis may only execute one program at a time.

For example, if motion program 1 is enabled to run, it must either complete or abort prior to enabling motion program 2.

Multiaxis Motion Programs and Sub-Routines

A motion program or sub-routine is designated multiaxis when it contains commands for both axis one and axis two. Both axes must be home referenced and meet the remaining prerequisites (see the section “Prerequisites for Programmed Motion” in this chapter) before a program can be executed. A multiaxis motion program may *CALL* only multiaxis sub-routines. One motion program instruction, *BLOCK-SYNC,* is available only in a multiaxis motion program. Sub-routine “nesting” limitations are the same as for a single axis motion program. In a multiaxis program, there are two categories of moves: 1-Axis moves and 2-Axis moves.

1-Axis moves: When two consecutive moves are programmed, the second move will begin execution within 2 milliseconds after the first move finishes.

2-Axis moves: A 2-Axis move is programmed with two consecutive blocks, one for each axis. The first of the two blocks must contain the BLOCK-SYNC command. When the BLOCK-SYNC command is executed, the two moves will be started “together” (within 2 milliseconds). Note that only the start of the moves is synchronized.

More information about multiaxis programming, program block structure, flow control (JUMP), and the *BLOCK-SYNC* command is provided later in this chapter.

Motion Program Command Types

The motion program commands are grouped into three categories:

Type 1 Commands

- Call Subroutine
- Jump

Type 2 Commands

- Block #
- Block SYNC
- Load Parameter
- Null
- Acceleration
- Velocity

Type 3 Commands

- Positioning Move
- Continuous Move
- Dwell
- Wait
- End of Program

Type 1 commands can redirect the program path execution, but do not directly affect positioning. Call Subroutine executes a subroutine before returning execution to the next command. Jumps may be conditional or unconditional. An unconditional jump always redirects execution to a specified

program location. A conditional jump is assigned a CTL bit to check. If the CTL bit is ON, the jump redirects execution to a specified program location. If the CTL bit is OFF, the jump is ignored.

Type 2 commands also do not affect position. The Block # command provides an identification or label for the Type 3 command that follows. If a program block does not contain a Block #, the previous Block # is used. The Block SYNC is a two-axis synchronization command (this may or may not delay motion on one axis). The Load Parameter command allows the user to load a value into a parameter register. The Velocity and Acceleration commands specify velocity and acceleration rates for the Type 3 MOVE command or commands that follow. Velocity and Acceleration commands remain in effect until changed.

Type 3 commands start or stop motion and thus affect positioning control. Positioning (PMOVE) and Continuous (CMOVE) moves command motion; the Dwell, Wait, and End of Program commands stop motion.

Program Blocks and Motion Command Processing

A "program block" consists of and is defined as one (and only one) Type 3 command with any number and combination of preceding Type 1 and 2 commands. The Block # command has two primary uses: (1) it provides a Jump-To identification (label), and (2) it identifies the section of the program that is currently executing via the two Command Block Number %AI Status words. Type 2 commands are optional; a program block can contain a single Type 3 command. Type 2 commands and Conditional Jumps do not take effect until the DSM executes the next Type 3 command.

While the DSM302 is executing a program block, the following program block is processed into a buffer command area. This buffering feature minimizes block transition time. Thus, parameters used in a move must be loaded before the move two blocks previous completes execution. In other words, in order to minimize the block-to-block transition time, a new block is pre-processed during previous block execution. Program block parameters must be loaded before the preceding block begins execution.

When a DSM302 is executing a 2-Axis program, the program commands are scanned independently by each axis and only the data designated for that axis is executed. Note that some commands do not specify an axis (Block #, Jump, Call, and End) and therefore apply to both axes.

A 2-Axis program can contain Sync Block # commands to synchronize the axes at designated points. When the first axis reaches the Sync Block, it will not execute the next block until the other axis has also reached the Sync Block. Refer to Example 18, Multiaxis Programming, later in this chapter, for an example of this.

Prerequisites for Programmed Motion

The following conditions must be satisfied before a motion program can be initiated (for a multiaxis program, the conditions must be met for BOTH axes):

- The *Enable Drive* %Q bit must be ON
- The *Drive Enabled* %I bit must be ON
- The *Position Valid* %I bit must be ON
- The *Moving* %I bit must be OFF

- The *Program Active* %I bit must be OFF
- The *Abort All Moves* %Q bit must be OFF
- The axis position must be within the configured end of travel limits (***Pos EOT*** and ***Neg EOT***), unless the ***Axis Mode*** is configured as CONTINUous
- The overtravel limit switches must be ON (24V input is high) if enabled
- A *Force Digital Servo Velocity* %AQ command must not be active
- The program to be executed must be a valid program stored in the DSM302

Conditions That Stop a Motion Program

A motion program will immediately cease when one of the following conditions occurs:

- The *Abort All Moves* %Q bit turns ON
- The *Enable Drive* %Q bit turns OFF
- An Overtravel Limit Switch turns OFF when ***OT Limit Switch*** is ENABLED via configuration.
- The next programmed move, either PMOVE or CMOVE, will pass a software end of travel limit (unless the axis mode is configured as Continuous)
- A Stop Normal or Stop Fast Response Method Error occurs. See Appendix A, “Error Reporting”

Parameters for Programmed Moves

Programmed moves have three parameters:

1. The *distance* (data) to move or *position* to move to,
2. The *type of positioning reference* (command modifier) to use for the move, and
3. The *type of acceleration* (command modifier) to use while performing the move.

Note: Motion programs can contain statements that use constants as data associated with commands or variables that are also referred to as parameters (P0-P255). (Parameter 0 is not supported by all configuration and programming software packages.)

Absolute or Incremental Positioning

Absolute Positioning

In an absolute positioning move, the first parameter is the position to move to. The following is an absolute positioning move example.

```
PMOVE 5000, ABS, LINEAR
```

In this example, the axis will move from its current position, whatever it may be, to the position 5000. Thus, the actual distance moved depends upon the axis' current position when the move is encountered. If the initial position is 0, the axis will move 5000 user units in the positive direction. If the initial position is 8000, the axis will move 3000 user units in the negative direction. If the initial position is 5000, the axis will not move.

Incremental Positioning

In an incremental move, the first parameter specifies the distance to move from the current position. The DSM302 translates incremental move distances into absolute move positions. This eliminates error accumulation. The following is an incremental positioning move example.

```
PMOVE 5000, INC, LINEAR
```

In this example, the axis will move from its current position to a position 5000 user units greater. With an incremental move, the first parameter specifies the actual number of user units the axis moves.

Types of Acceleration

Linear Acceleration

A sample linear move profile that plots velocity versus time is shown in Figure 7-1. As illustrated, a linear move uses constant (linear) acceleration. The area under the graph represents the distance moved.

```
ACCEL  1000
VELOC 2000
PMOVE  6000, INC, LINEAR
```

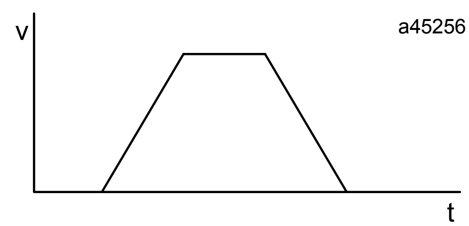


Figure 7-1. Sample Linear Motion

S-Curve Acceleration

An S-Curve motion sample, plotting velocity versus time, is shown below. As illustrated, S-Curve acceleration is non-linear. When the move begins, the acceleration starts slowly and builds until it reaches the programmed acceleration. This should be the midpoint of the acceleration. Then, the acceleration begins decreasing until it is zero, at which time the programmed velocity has been reached. An S-Curve move requires twice the time and distance to accelerate and decelerate that a comparable linear move needs. The area under the graph represents the distance moved.

```
ACCEL    2000
VELOC    2000
PMOVE    8000, INC, SCURVE
```

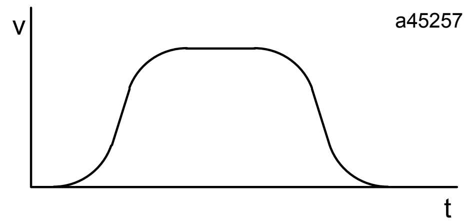


Figure 7-2. Sample S-Curve Motion

Types of Programmed Move Commands

Positioning Move (PMOVE)

A PMOVE must always come to a complete stop. The stop must long enough to allow the *In Zone* %I bit to turn ON before the next move can begin.

A PMOVE uses the most recently programmed velocity and acceleration. If a VELOC command has not been encountered in the motion program, the ***Jog Velocity*** is used as default. If an ACCEL command has not been encountered in the motion program, the ***Jog Acceleration*** is used as default.

Continuous Move (CMOVE)

A CMOVE does not stop when completed unless it is followed by a DWELL or a WAIT, the next programmed velocity is zero, or it is the last program command. It does not wait for *In Zone* %I bit to turn ON before going to the next move. <u>A normal CMOVE is a command that reaches its programmed position at the same time that it reaches the velocity of the *following* Move command.</u>

A CMOVE uses the most recently programmed velocity and acceleration. If a VELOC command has not been encountered in the motion program, the ***Jog Velocity*** is used as default. If an ACCEL command has not been encountered in the motion program, the ***Jog Acceleration*** is used as default.

A special form of the CMOVE command can be used to force the DSM302 to reach the programmed CMOVE position *before* starting the velocity change associated with the *next* move command (that is, execute the entire CMOVE command at a constant velocity). Programming an incremental CMOVE command with an operand of 0 (CMOVE INC 0) will force a delay in the servo velocity change for the *next* move command in sequence. The following sequence of commands illustrates this effect (assume ACCELs are chosen to allow motions to complete normally):

Command	Data	Comments
VELOC	10000	Set velocity of first move = 10000
CMOVE	15000, ABS, LINEAR	Reach velocity of second move (20000) at position = 15000
VELOC	20000	Set velocity of second move = 20000
CMOVE	0, INC, LINEAR	Flag to signal the DSM302 to wait for next move before changing to the next velocity
CMOVE	30000, ABS, LINEAR	Stay at velocity = 20000 until position = 30000, then change to velocity = 5000
VELOC	5000	Set velocity of third move = 5000
PMOVE	40000, ABS, LINEAR	Final stop position = 40000

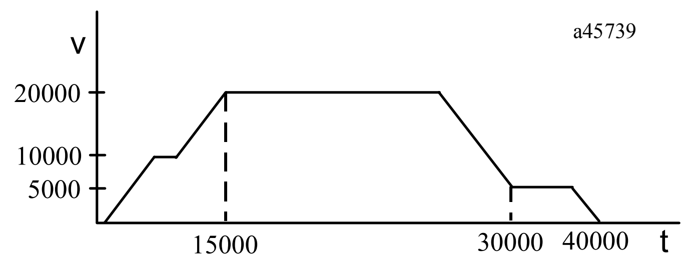


Figure 7-3. Example 1, Before Inserting CMOVE (0)

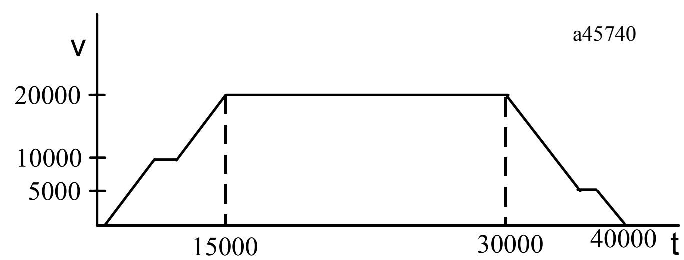


Figure 7-4. Example 2, After Inserting CMOVE (0)

Programmed Moves

By combining CMOVEs and PMOVES, absolute and incremental moves, and linear and s-curve motion, virtually any motion profile can be generated. The following examples show some simple motion profiles, as well as some common motion programming errors.

Example 1: Combining PMOVEs and CMOVEs

This example shows how simple PMOVEs and CMOVEs combine to form motion profiles.

```
ACCEL  1000
VELOC 2000
PMOVE 5000, ABS, LINEAR
VELOC 1200
PMOVE 10000, ABS, SCURVE
ACCEL  1500
VELOC 2800
CMOVE  6000, INC, LINEAR
VELOC 1200
CMOVE  23000, ABS, SCURVE
ACCEL  1000
VELOC 2800
PMOVE 5000, INC, LINEAR
```

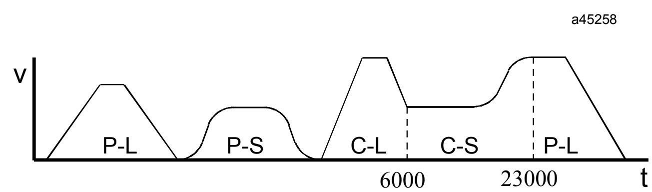


Figure 7-5. Combining PMOVEs and CMOVEs

The move types are indicated under the corresponding move; for example, P-L indicates linear PMOVE.

The first PMOVE accelerates to program velocity, moves for a distance, and decelerates to a stop. This is because motion stops after all PMOVEs. When the first move stops, it is at the programmed distance.

The second move is an s-curve PMOVE. It, like the first, accelerates to the programmed velocity, moves for a time, and decelerates to zero velocity because it is a PMOVE.

The next move is a linear CMOVE. It accelerates to program velocity, moves for a time, and then decelerates to a lower velocity using linear acceleration. When a CMOVE ends, it will be at the programmed position of the move just completed, and at the velocity of the next move. Thus when the fourth move begins, it is already at its programmed velocity.

The fourth move is a CMOVE, so as it approaches its final position, it accelerates to be at the velocity of the fifth move when it completes. The graph shows the acceleration of the fourth move is s-curve.

Finally, the fifth move begins and moves at its programmed velocity for a time until it decelerates to zero. Any subsequent moves after the fifth would begin at zero velocity because the fifth move is a PMOVE.

Example 2: Changing the Acceleration Mode During a Profile

The following example shows how a different acceleration, and an even acceleration mode, can be used during a profile using CMOVEs. The first CMOVE accelerates linearly to the programmed velocity. Because the second CMOVE's velocity is identical to the first, the first CMOVE finishes its move without changing velocity. The acceleration of the second move is S-curve as it decelerates to zero velocity.

```
ACCEL    2000
VELOC    6000
CMOVE    13000, ABS, LINEAR
ACCEL    4000
CMOVE    15000, INC, SCURVE
```

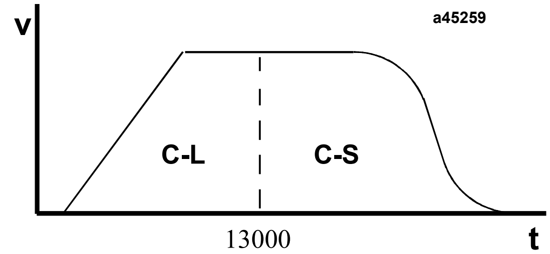


Figure 7-6. Changing the Acceleration Mode During a Profile

Example 3: Not Enough Distance to Reach Programmed Velocity

CMOVES and PMOVES can be programmed that do not have enough distance to reach the programmed velocity. The following graph shows a CMOVE that could not reach the programmed velocity. The DSM302 accelerates to the point where it must start decelerating to reach the programmed position of C1 at the velocity of the second CMOVE.

```
ACCEL 2000
VELOC 8000
CMOVE  7000, INC, LINEAR
ACCEL 10000
VELOC 2000
CMOVE  4400, INC, LINEAR
```

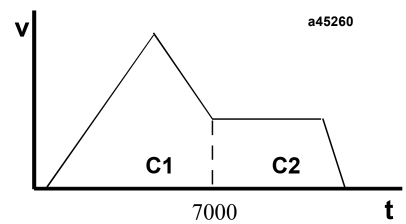


Figure 7-7. Not Enough Distance to Reach Programmed Velocity

Example 4: Hanging the Move When the Distance Runs Out

A serious programming error involves "hanging" (i.e. leaving no desirable options for the command generator) the move at a high velocity when the distance runs out. In the following example, the first CMOVE accelerates to a high velocity. The second CMOVE has an identical velocity. However, the distance specified for the second CMOVE is very short. Thus, the axis is running at a very high velocity and must stop in a short distance. If the programmed acceleration is not large

enough, the following profile could occur. The DSM302 attempts to avoid overshooting the final position by commanding a zero velocity. This rapid velocity change is undesirable and can cause machine damage.

```
ACCEL   500
VELOC   3000
CMOVE   9000, ABS, LINEAR
ACCEL   600
CMOVE   4800, INC, LINEAR
```

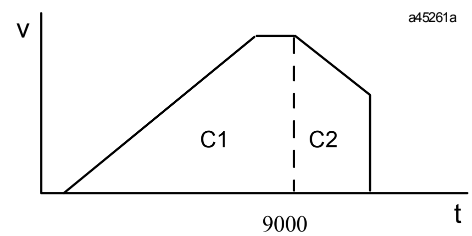


Figure 7-8. Hanging the DSM302 When the Distance Runs Out

Dwell Command

A DWELL command is used to generate no motion for a specified number of milliseconds. A DWELL after a CMOVE will make the CMOVE perform similar to a PMOVE, even if the specified dwell duration is zero milliseconds.

A DWELL-P command uses the value stored in the designated parameter. It is treated as a NULL command and skipped (CMOVE continues to the next Move following the DWELL-P) if the parameter value has a special “flag” value of 65000. (This does not occur with a DWELL command that has a constant value of 65000.)

Example 5: Dwell

A simple motion profile, which moves to a specific point, waits, and returns to the original point is shown below.

```
ACCEL  30000
VELOC  15000
PMOVE  120000, ABS, LINEAR
DWELL  4000
PMOVE  0, ABS, LINEAR
```

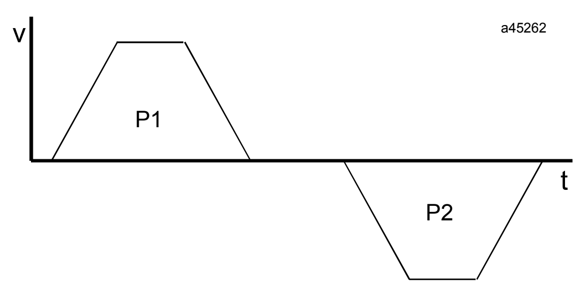


Figure 7-9. Dwell Command Example

Wait Command

The WAIT command is similar to the DWELL command. Instead of generating no motion for a specified period of time, a WAIT stops program motion until a specified CTL bit turns ON. Thus motion stops any time a WAIT is encountered, even if the CTL bit is ON before the WAIT is reached in the program. The trigger to continue the program can be any of the twelve CTL bits.

If, in the previous example, WAIT was substituted for DWELL, the motion profile would be the same except the second PMOVE would not start until the CTL bit turned ON. If the CTL bit was ON when the program reached the WAIT, the second PMOVE would begin immediately after the first PMOVE finished.

Also, if WAIT was used instead of DWELL in the previous example, CMOVEs and PMOVEs would generate similar velocity profiles. The WAIT will stop motion whether the previous move is a CMOVE or PMOVE.

Subroutines

The DSM302 can store up to ten separate programs and forty subroutines. Subroutines can be defined as two types: *single axis* and *multiaxis*. Subroutines are available for all motion programs created with the Motion Programmer software. However, Program 0 in the configuration software does not support subroutines. Commands within single axis subroutines do not contain an axis number; this allows single axis subroutines to be called from any single axis program written for either axis 1 or axis 2. Commands within multiaxis subroutines contain axis numbers just like commands within multiaxis programs. Multiaxis subroutines can only be called from multiaxis programs or subroutines. Single axis subroutines can only be called from single axis programs or subroutines. A single axis program for axis 1 and a single axis program for axis 2 can call the same single axis subroutine simultaneously. Each subroutine must be assigned a unique number between 1 and 40.

Subroutines are programmed using the CALL command which specifies the subroutine number to be called. When a CALL is encountered during program execution, program execution is redirected to the subroutine. When the subroutine completes, program execution resumes at the command after the CALL command. Subroutines can be called from another subroutine, but once a subroutine has been called, it must complete before it can be called again for the same axis. Thus, recursion is not allowed.

Block Numbers and Jumps

Block numbers are used as reference points within a motion program and to control jump testing. A %AI data word displays the current block number which can be monitored to ensure correct program execution or to determine when events should occur. A block number can also serve as a JUMP command destination. Jumps are divided into unconditional and conditional. An unconditional jump command simply tells the DSM302 to continue program execution at the destination block number. A conditional jump only executes if the specified condition occurs. Examples of both types of jumps follow.

Unconditional Jumps

Example 6: Unconditional Jump

In the example below, the program executes a PMOVE, dwells for 2 seconds, then unconditionally jumps back to the beginning of the program at block 1. Thus, the PMOVE repeats until an end of travel limit (***Pos EOT*** or ***Neg EOT***) or Overtravel Limit Switch is reached. A more practical application would be to program a series of moves that needed to be repeated continuously, then

add a jump back to just before the first move to close the loop. In that case, the program would continue to repeat the series of moves until a %Q or %AQ command from the ladder program stops or alters it.

```
ACCEL  10000
VELOC  30000
BLOCK  1
PMOVE  200000, INC, LINEAR
DWELL  2000
JUMP   UNCOND, 1
```

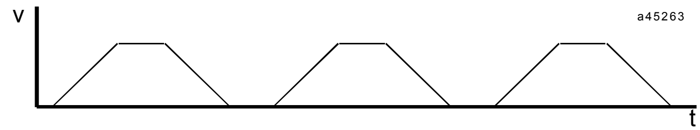


Figure 7-10. Unconditional Jump

Conditional Jumps

A conditional jump is a JUMP command with a CTL bit specified in the command. Conditional jumps are Type 1 commands in that they affect program path execution, but they are also similar to Type 2 commands because they do not take effect until a Type 3 command following the JUMP command is executed. When a conditional JUMP command is executed, the DSM302 examines the specified CTL bit. If the bit is ON, program execution jumps to the destination BLOCK #; if the bit is OFF, the program continues executing the command after the JUMP. Note that the Type 3 command after the conditional jump and at the jump destination will affect jump behavior.

Conditional JUMP commands should <u>not</u> be used with multiaxis programs containing sync blocks unless the Jump is triggered while both axes are testing the same JUMP command. Failure to follow this recommendation can result in unpredictable operation.

Conditional Jump testing **starts** when the next PMOVE, CMOVE, DWELL, WAIT or END Program command following a Conditional JUMP becomes active.

When Conditional Jump testing is active, the designated CTL bit is tested once every 2 milliseconds.

Conditional Jump testing **ends** when the designated CTL bit turns ON (Jump Trigger occurs) or when a new Block Number becomes active.

If more than one Conditional JUMP is programmed without an intervening PMOVE, CMOVE, DWELL, WAIT or END Program command, only the last Conditional JUMP will be recognized.

In summary, a Conditional JUMP transfers control to a new program block on the basis of one of the external CTL input bits turning ON. Tests for CTL bit status can be carried out once or continuously during the following Type 3 command if it is in the same program block. Multiple conditional jumps are not supported within the same program block.

Motion Program Example 1:

Begin Program		
BLOCK	1	
JUMP	CTL01, 2	This JUMP command will be ignored
JUMP	CTL02, 3	This JUMP command will be recognized
CMOVE	1, +40000, INC, LINEAR	
BLOCK	2	
CMOVE	1, +20000, INC, LINEAR	
BLOCK	3	
PMOVE	1, +100000, ABS, LINEAR	
BLOCK	4	
DWELL	1, 100	
End Program		

When a new Block Number becomes active AFTER a Conditional JUMP command, Jump testing will occur one final time.

Motion Program Example 2:

Begin Program		
BLOCK	1	
CMOVE	1, +20000, ABS, LINEAR	
JUMP	CTL01, 3	
BLOCK	2	
PMOVE	1, +40000, ABS, LINEAR	CTL01 tested only once
BLOCK	3	
DWELL	1, 100	
End Program		

In the example above, The *CTL01* bit test occurs just once because the PMOVE following the JUMP contains a new Block Number (2).

Changing the location of Block Number 2 causes CTL bit testing throughout the PMOVE following the JUMP:

Motion Program Example 3:

Begin Program		
BLOCK	1	
CMOVE	1, +20000, ABS, LINEAR	
BLOCK	2	
JUMP	CTL01, 3	
PMOVE	1, +40000, ABS, LINEAR	CTL01 tested throughout PMOVE
BLOCK	3	
DWELL	1, 100	
End Program		

The DSM302 can perform a Conditional JUMP from an active CMOVE to a program block containing a CMOVE or PMOVE without stopping. **For the axis to jump without stopping, the distance represented by the CMOVE or PMOVE in the Jump block must be greater than the servo stopping distance.** The servo stopping distance is computed using the present commanded velocity and the acceleration parameters that would be in effect when the jump block became active.

The axis will STOP before jumping if a Conditional Jump trigger occurs under any of the following conditions:

- When a PMOVE is active
- When a CMOVE is active and the Jump destination block contains a CMOVE or PMOVE representing motion in the opposite direction.
- When a CMOVE is active and the Jump destination block contains a CMOVE or PMOVE representing motion in the same direction with insufficient distance for the axis to stop.
- When a CMOVE is active and the Jump destination block contains a DWELL, WAIT or END (program) command.

If the axis does STOP before a Conditional Jump, the ***Jog Acceleration*** and ***Jog Acceleration Mode*** will be used.

Unconditional Jumps do not force the axis to stop before jumping to a new program block. For example, a CMOVE followed by a JUMP Unconditional to another CMOVE will behave just as if the two CMOVEs occurred without an intervening Unconditional JUMP.

If Conditional Jump testing is active when the DSM302 command processor encounters a CALL SUBROUTINE command, the axis will **stop** and terminate jump testing before the CALL is executed.

If Conditional Jump testing is active when the DSM302 command processor encounters an END SUBROUTINE command, the axis will **stop** and terminate jump testing before the END SUBROUTINE is executed.

Jump Testing

Conditional jumps perform jump testing. If the CTL bit is ON, the jump is immediately performed. If the CTL bit is OFF, the DSM302 watches the CTL bit and keeps track of the JUMP destination. This monitoring of the CTL bit is called jump testing. If during jump testing the CTL bit turns ON before a BLOCK command, another JUMP command, or a CALL command is encountered, the jump is performed. These commands will end jump testing.

Example 7: Jump Testing

Consider the following two program section examples. In Example 1, the move to position 2000 is completed before jump testing begins. The BLOCK command occurring immediately after the JUMP command ends jump testing. Thus the duration for which the CTL bit is monitored is very short. However, in Example 2, the JUMP command is encountered before the move command. This starts the jump testing before motion begins, and jump testing continues as long as the move lasts. If the CTL bit turns ON while the move is being performed, the jump will be performed. After the move completes, the BLOCK command ends jump testing and program execution

continues normally. Jump testing would continue during subsequent moves encountered before the BLOCK command.

Example 1	**Example 2**
ACCEL 5000	ACCEL 5000
VELOC 1000	VELOC 1000
BLOCK 1	BLOCK 1
CMOVE 2000, ABS, LINEAR	JUMP CTL01, 3
JUMP CTL01, 3	CMOVE 2000, ABS, LINEAR
BLOCK 2	BLOCK 2

Normal Stop Before JUMP

A conditional jump command is similar to Type 2 commands in that jump testing does not start until the Type 3 command immediately after the JUMP is executed. If this Type 3 command would normally stop motion, then motion will stop before jump testing begins. Type 3 commands that will stop motion are: DWELL, WAIT, End of Program, and moves in the opposite direction.

Thus even though the CTL bit may be ON before the block with the conditional JUMP and Type 3 command is executed, axis motion will stop before program execution continues at the jump destination. This stopping is NOT a Jump Stop, which is described in Example 10.

Example 8: Normal Stop Before JUMP

The following example contains a jump followed by a DWELL command. The DSM302, because it processes ahead, knows it must stop after the CMOVE command. Thus, it comes to a stop before the DWELL is executed. Since jump testing does not begin until the DWELL is executed, testing begins after motion stops. Jump testing ends when the following CMOVE begins due to the associated BLOCK command. The dashed lines in the velocity profile indicate when jump testing takes place. In this example, the CTL03 bit does not turn ON during the program execution.

```
BLOCK 1
ACCEL  5000
VELOC 10000
CMOVE  60000, INC, LINEAR
BLOCK 2
JUMP    CTL03, 4
DWELL 4000
BLOCK 3
ACCEL  10000
VELOC 5000
CMOVE  15000, INC, LINEAR
BLOCK 4
NULL
```

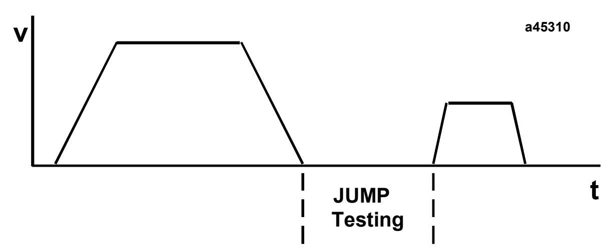


Figure 7-11. Normal Stop Before JUMP

Jumping Without Stopping

If the Type 3 command following a conditional jump is a CMOVE and the Type 3 command at the destination is a move command with sufficient distance to fully decelerate to zero when completed, the jump will be executed without stopping. This is the only way to sustain motion when a jump is performed.

Example 9: JUMP Without Stopping

This is a simple example of a conditional jump from one CMOVE to another. While jump testing the *CTL03* bit, the first CMOVE accelerates to the programmed velocity. Before the dashed line, the *CTL03* bit is OFF, but at the dashed line the *CTL03* bit turns ON. Program execution is immediately transferred to block 3 and the CMOVE there begins. Because the velocity at the jump destination is different, the velocity changes at the acceleration programmed of the jump destination block. Finally, as the second CMOVE completes, velocity is reduced to zero and the program ends.

```
BLOCK 1
ACCEL  2000
VELOC  10000
JUMP    CTL03, 3
CMOVE  120000, INC, LINEAR
BLOCK 3
ACCEL  20000
VELOC  5000
CMOVE  15000, INC, LINEAR
```

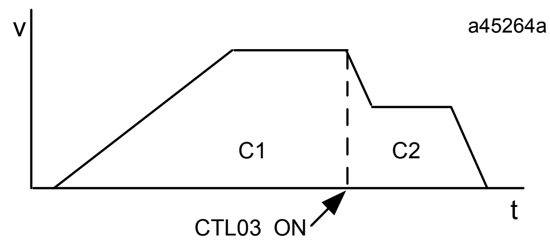


Figure 7-12. JUMP Without Stopping

Jump Stop

A jump stop is a stop that is caused by a jump. When a jump stop occurs, the ***Jog Acceleration*** and ***Jog Acceleration Mode*** are used instead of any programmed acceleration. Note that s-curve motion will achieve constant velocity before using the ***Jog Acceleration*** and beginning to decelerate. See the s-curve jump examples for more details. The ***Jog Acceleration*** is used because a jump stop may indicate something is wrong. The current ***Jog Acceleration***, which can be changed with an immediate command, provides more versatility than the programmed acceleration. There are two ways of generating a jump stop each described below.

A conditional JUMP triggered during a PMOVE will always generate a jump stop. Because a PMOVE always stops before continuing to a subsequent motion, a jump stop always occurs when a jump takes place during a PMOVE.

When a conditional jump trigger occurs during a CMOVE, however, a jump stop will not occur if the motion programmed at the jump destination is a PMOVE or CMOVE representing sufficient distance in the same direction. A jump stop will occur if the PMOVE or CMOVE at the jump destination does not represent sufficient distance or represents motion in the opposite direction.

In an s-curve move, a jump stop will do one of two things. If the jump takes place after the midpoint of the acceleration or deceleration, the acceleration or deceleration is completed before the jump stop is initiated. If the jump occurs before the midpoint of the acceleration or

deceleration, the profile will immediately begin leveling off. Once acceleration or deceleration is zero, the jump stop begins. See the s-curve jump examples.

Example 10: Jump Stop

The following is an example conditional jump with a jump stop. An enhancement on Example 5, DWELL, would be to watch an external CTL bit that would indicate a problem with the positive motion. If the CTL bit never turns on, the profile for the following program will be identical to the profile shown in the DWELL example. If the CTL bit turned on during the first PMOVE or the DWELL, the reverse movement would immediately commence.

The following profile would appear if the CTL bit turned on during the first PMOVE, at the dashed line, and the *Jog Acceleration* was 75000. Because the first move completed early due to the CTL bit and a faster acceleration (***Jog Acceleration*** versus programmed acceleration) the second move would not have to move as far to get back to 0 position as it did in the DWELL example. Note that because the motion programmed at the jump destination is in the opposite direction as the initial motion, the profile would be identical if the moves were CMOVEs instead of PMOVEs.

```
ACCEL     30000
VELOC     15000
BLOCK     1
JUMP      CTL09, 2
PMOVE     120000, ABS, LINEAR
DWELL     4000
BLOCK     2
PMOVE     0, ABS, LINEAR
```

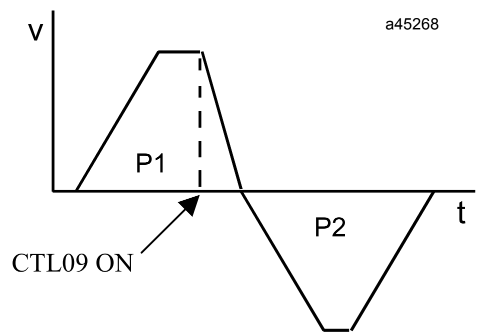


Figure 7-13. Jump Stop

Example 11: Jump Followed by PMOVE

In this JUMP example, the command after the JUMP is a PMOVE in the same direction. The velocity profile below shows the acceleration and movement for the first CMOVE and the deceleration to the PMOVE's velocity. The *CTL01* bit, OFF when the PMOVE begins, turns ON at the second dashed line. Motion stops after a PMOVE, even if a conditional jump goes to another block. Thus the *CTL01* bit triggers a deceleration to zero before the final CMOVE begins.

```
BLOCK 1
ACCEL  2000
VELOC 8000
CMOVE  76000, INC,
LINEAR
BLOCK 2
ACCEL  1000
VELOC 4000
JUMP    CTL01, 3
PMOVE 50000, INC, LINEAR
BLOCK 3
ACCEL  6000
VELOC 6000
CMOVE  6000, INC, LINEAR
```

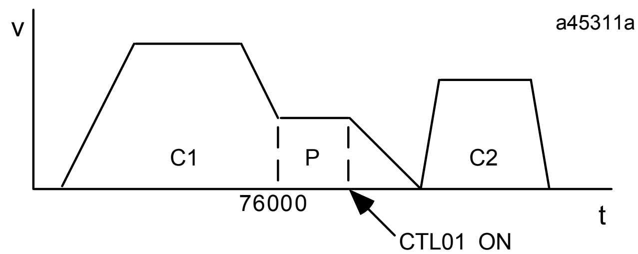


Figure 7-14. Jump Followed by PMOVE

S-CURVE Jumps

Jumps during linear motion and jumps during s-curve motion at constant velocities immediately begin accelerating or decelerating to a new velocity. Jumps during a s-curve acceleration or deceleration, however, require different rules in order to maintain a s-curve profile. What happens when a jump occurs during an s-curve move while changing velocity depends on whether the jump occurs before or after the midpoint (the point where the acceleration magnitude is greatest) and whether the velocity at the jump destination is higher or lower than the current velocity.

S-CURVE Jumps after the Midpoint of Acceleration or Deceleration

If the jump occurs after the midpoint of the change in velocity, the change will continue normally until constant velocity is reached; then the velocity will be changed to the new velocity using the acceleration mode of the move at the jump destination.

Example 12: S-CURVE - Jumping After the Midpoint of Acceleration or Deceleration

In the following example, a jump occurs during the final phase of deceleration, at the dashed line. The deceleration continues until constant velocity is reached and then the acceleration to the higher velocity begins.

ACCEL 50000
VELOC 100000
BLOCK 1
JUMP CTL01, 3
CMOVE 500000, ABS, SCURVE
BLOCK 2
VELOC 60000
CMOVE 500000, INC, SCURVE
BLOCK 3
VELOC 85000
ACCEL 100000
CMOVE 250000, INC, SCURVE

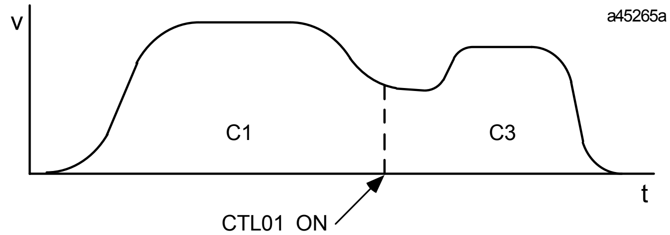


Figure 7-15. Jumping After the Midpoint of Acceleration or Deceleration

S-CURVE Jumps before the Midpoint of Acceleration or Deceleration

If a jump takes place before the midpoint of acceleration or deceleration, the result depends on whether the velocity at the jump destination is higher or lower than the velocity before the jump took place. In the first case, when accelerating but the new velocity is lower, or decelerating and the new velocity is greater, the DSM302 will immediately begin reducing the acceleration or deceleration to zero. Once at zero velocity, the DSM302 will use the jump destination acceleration and velocity and change to the new velocity.

Example 13: S-CURVE - Jumping Before the Midpoint of Acceleration or Deceleration

In the following example, during the acceleration of the first CMOVE, a jump takes place at the first dashed line. Because the velocity at the jump destination is lower than the velocity of the first CMOVE the DSM302 slows the acceleration to zero. Constant velocity, zero acceleration, occurs at the second dashed line. There, the DSM302 begins decelerating to the new velocity using the acceleration at the jump destination. Finally, the second CMOVE finishes.

ACCEL 1000
VELOC 50000
BLOCK 1
JUMP CTL01, 3
CMOVE 50000, INC, SCURVE
BLOCK 3
VELOC 5000
ACCEL 10000
CMOVE 15000, INC, SCURVE

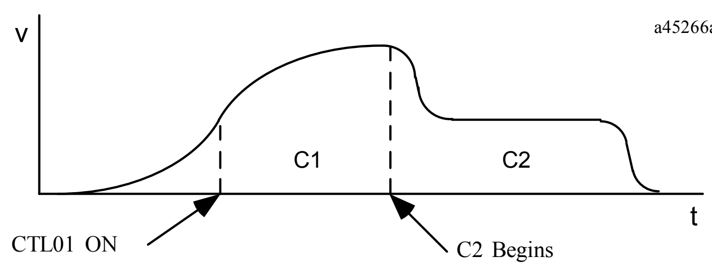


Figure 7-16. Jumping before the Midpoint of Acceleration or Deceleration

S-CURVE Jumps to a higher Acceleration while Accelerating or a lower Deceleration while Decelerating

The second case involves jumping to a higher velocity while accelerating or a lower velocity while decelerating. When this occurs, the DSM302 continues to the first move's acceleration or deceleration. This acceleration or deceleration is maintained, similar to be a linear acceleration, until the axis approaches the new velocity. Then the normal S-curve is used to reduce acceleration or deceleration to zero.

Example 14: S-CURVE - Jumping to a Higher Velocity While Accelerating or Jumping to a Lower Velocity While Decelerating

In this example, a JUMP command is triggered during the initial phase of acceleration (at the first dashed line) and the velocity at the jump destination is higher than that of the current move. The first dashed line indicates the maximum acceleration of the first CMOVE. This value is held as the axis continues to accelerate until it s-curves back to constant velocity. Constant velocity, the second dashed line, indicates the beginning of the second CMOVE. This move continues until it decelerates to zero at the end of the program.

```
ACCEL  50000
VELOC 30000
BLOCK 1
JUMP   CTL02, 2
CMOVE  150000, INC, SCURVE
BLOCK 2
VELOC 90000
ACCEL  25000
CMOVE  500000, INC,
SCURVE
```

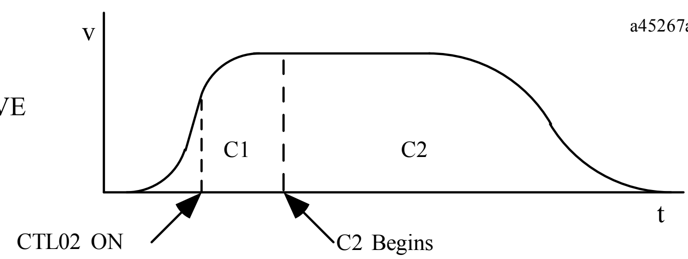


Figure 7-17. Jumping to a Higher Velocity While Accelerating or Jumping to a Lower Velocity While Decelerating

Other Programmed Motion Considerations

Maximum Acceleration Time

The maximum time for a programmed acceleration or deceleration is 131 seconds. If the time to accelerate or decelerate is computed to be longer than this time, the DSM302 will compute an acceleration to be used based on 131 seconds. To obtain longer acceleration times, multiple CMOVEs with increasing or decreasing velocities must be used.

Example 15: Maximum Acceleration Time

The following two program examples show a hypothetical problem with a very long acceleration time in Example1, and a possible solution in Example 2. In Example 1 below, 240 seconds is required to reach the programmed velocity of 24,000 at an acceleration rate of 100 (24000 ÷ 100 = 240). Since this is greater than the DSM's limit of 131 seconds per acceleration or deceleration, the DSM will calculate a value within its limit. In this case, the DSM calculates that to reach a velocity of 24,000 in 131 seconds, an acceleration of 183 would be required. The Example 1 solid line velocity profile shows the higher (183) acceleration rate used by the DSM. The dashed line profile in that drawing indicates the desired (programmed) acceleration rate and velocity profile that could not be attained.

```
ACCEL  100
VELOC  24000
PMOVE  8000000, INC, LINEAR
```

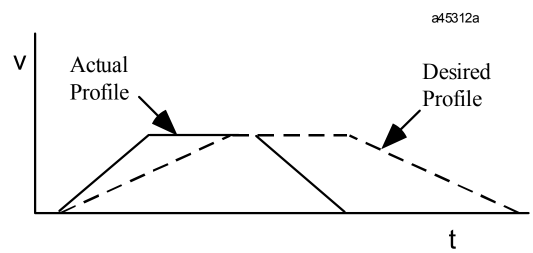


Figure 7-18. Maximum Acceleration Time Example 1

One solution (which requires some extra calculations) for obtaining a low acceleration for a long period of time breaks a move up into separate continuous moves (using CMOVE commands), with each move's acceleration time being less than 131 seconds. In the problem introduced in Example 1, the programmed move would require 240 seconds each for acceleration and deceleration. We can easily see that if we divide this time in half, by using two moves whose acceleration or deceleration times are each 120 seconds, we would be within the DSM's limit of 131 seconds. This scheme is used in the following example.

Example 2 below shows how the result desired in Example 1 could be obtained by replacing Example 1's single move with four moves. Four moves are required since both the acceleration and deceleration portions of the profile must each be divided into two moves. To divide the total acceleration (or deceleration) time in half, we calculate the distance at the midpoint of either slope, when velocity is 12000, to be 720,000 user units.

The distance traveled during acceleration or deceleration is calculated using the formula:

$$\text{Distance traveled} = \frac{\text{Change in velocity x Required time}}{2}$$

$$720{,}000 = \frac{12{,}000 \times 120}{2}$$

(Since 240 seconds is needed to reach a velocity of 24,000, a velocity of 12,000 can be reached in 120 seconds.) The initial CMOVE and the final PMOVE both use this distance. A second CMOVE “takes over” at the midpoint of the acceleration slope from the first CMOVE and accelerates to the target velocity of 24,000. A third CMOVE is required for dividing up the deceleration portion of the profile. The final move, a PMOVE, “takes over” from the third CMOVE at the deceleration midpoint distance (720,000 user units from the final position). The third CMOVE, as it approaches its final position, will automatically decelerate to the PMOVE’s velocity of 12,000. The dashed lines in the Example 2 drawing separate the four moves. To calculate the distances of the second and third CMOVEs, we subtracted the distances we calculated for the first CMOVE and final PMOVE (720,000 each for a total of 1,440,000) from the final distance of 8,000,000. This gave us a remaining distance of 6,560,000, which we divided equally between the second and third CMOVES (3,280,000 each).

```
ACCEL   100
VELOC   12000
CMOVE  720000, INC, LINEAR
VELOC   12000
CMOVE  3280000, INC, LINEAR
VELOC   24000
CMOVE  3280000, INC, LINEAR
VELOC   12000
PMOVE   720000, INC, LINEAR
```

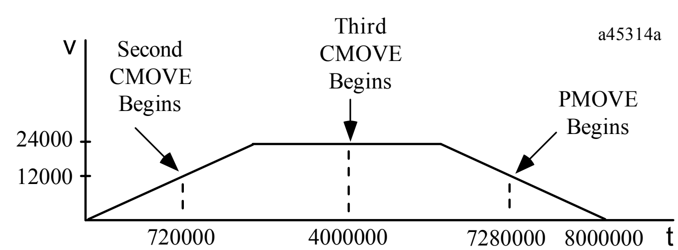


Figure 7-17. Maximum Acceleration Time Example 2

Feedhold with the DSM302

Feedhold is used to temporarily pause program execution without ending the program, often to examine some aspect of a system. It causes all axis motion to end at the programmed acceleration. When Feedhold is ended, program execution resumes. Interrupted motion will resume at the programmed acceleration and velocity.

Feedhold is asserted by turning ON the *Feed Hold* %Q bit and lasts until the %Q bit is turned OFF. The *Abort All Moves* %Q bit turning ON or an error that would normally cause a stop error will end feedhold as well as terminate the program. During Feedhold, jogging positive and negative is allowed, but no other motion. When Feedhold is terminated and program execution resumes, the DSM302 remembers and will move to its previous destination.

Example 16: Feedhold

The following example illustrates a motion profile when Feedhold is applied. The linear move accelerates to the programmed velocity at the programmed rate. Feedhold is applied at the dashed line, so velocity decreases at the programmed acceleration to zero. Then, a Jog is performed using the *Jog Minus %Q* bit. This is evident because the jog velocity is negative. Note that the acceleration used during the Jog is the current ***Jog Acceleration,*** which is different than the programmed acceleration. Note also, the *Feed Hold* %Q command must be applied during the entire duration of the Jog. After the jog motion has ceased, the Feedhold is ended and the program continues to completion.

ACCEL 1000
VELOC 2000
PMOVE 12000, INC, LINEAR

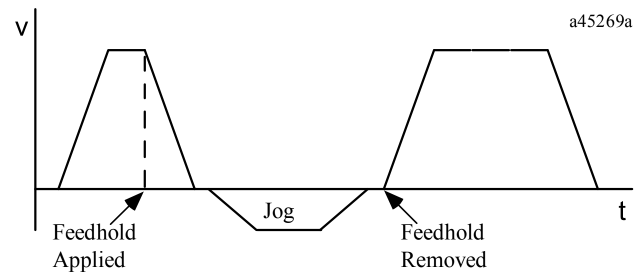


Figure 7-19. Feedhold Example

Feedrate Override

Some applications require small modifications to a programmed velocity to handle outside changes. A *Rate Override* %AQ immediate command, which is sent to the DSM through ladder logic, allows changes to a programmed feedrate (velocity) during program execution. (Details about the Rate Override command are found in Chapter 5.) When a program begins executing, the override rate is initially set to 100%. Thus, changes to feedrate before the execute program bit is turned ON will be ignored. However, a rate override commanded on the same sweep as an execute program bit *will* be effective.

A percentage can be assigned to the feedrate override of from 0% to 120%. When a *Rate Override* is commanded, the DSM302 internally multiplies the feedrate percentage by programmed velocity to obtain a new velocity. If the axis is moving, the current move's ***Jog Acceleration Mode*** is used to change velocity to the new velocity. All future move velocities will be affected by the feedrate change. Note that when a feedrate of 0% is applied, no motion will be generated until a new feedrate is commanded. Also note the *Moving %I* bit stays ON when the feedrate is 0%.

Rate Override has no effect on non-programmed motion such as *Jog, Find Home,* or *Move at Velocity.*

Example 17: Feedrate Override

During execution of this program, feedrate changes of + or -10% are commanded. Dotted lines indicate -10%, dashed lines indicate +10%.

ACCEL	1000
VELOC	6000
PMOVE	110000, INC, LINEAR

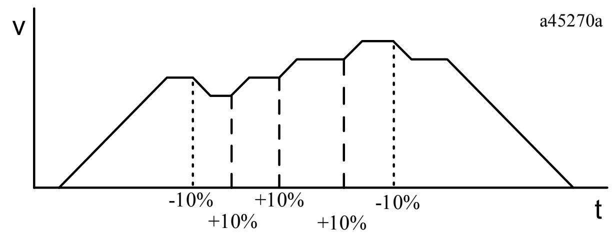


Figure 7-20. Feedrate Override Example

Multiaxis Programming

Sync Blocks can be used in a multiaxis program to synchronize the axis motion commands at positions where timing is critical.

Example 18: Multiaxis Programming

This example assumes that axis 1 controls vertical motion and axis 2 controls horizontal motion. The objective is to move a piece of material from point A to point E as quickly as possible while avoiding the obstacle that prevents a direct move between those points.

A simple way would be to move straight up from point A to point C, and then from point C to point E. This sequence, however, wastes time. A better way would begin the horizontal movement before reaching point C. It has been determined that after axis 1 has moved to a position of 30,000, user units (to point B), axis 2 could then start and still clear the obstacle. The program segment could be programmed as follows:

BLOCK 10	CMOVE, 30000, INC	AXIS 1
BLOCK 20 [SYNC]	PMOVE, 50000, INC	AXIS 1
	PMOVE, 120000, INC	AXIS 2

When BLOCK 10 is executed, axis 1 begins its 30,000-unit move while axis 2 pauses. When the axis 1 move completes, two things occur: axis 1 begins the 50,000-unit PMOVE commanded in BLOCK 20 (SYNC) without stopping (because the first move was a CMOVE), and axis 2 begins its 120,000-unit move. In the figure below, the axis 1 first move transfers the part from point A to point B. At point B, axis 1 continues moving (performing its second move) and axis 2 begins its move, bringing the part to point D. Axis 1 completes its second move at point D and stops; however, axis 2 continues, and moves the part to point E.

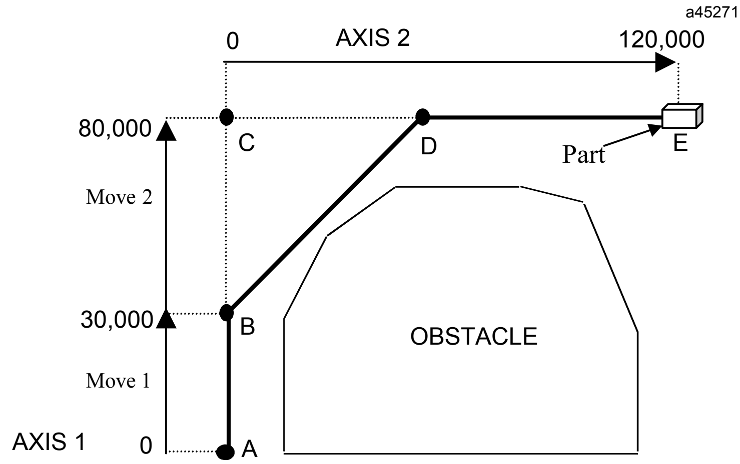


Figure 7-21. Multiaxis Programming Example

If this program segment is not at the beginning of a program, and for some reason axis 2 has not yet reached Block 20 when axis 1 has moved 30,000 counts, an error would occur. Axis 1 would continue to 80,000 counts, and the DSM302 would report a "Block Sync Error during a CMOVE" in the Status Code.

If it is imperative that the axes synchronize at Block 20, changing Block 10 to a PMOVE would guarantee synchronization, but then axis 1 would stop at 30,000 counts.

Parameters (P0-P255) in the DSM302

The DSM302 maintains 256 double word parameters (0 through 255) in memory. (Note that parameter 0 is not supported in all configuration or motion program software packages.) These values can be used as a parameter in ACCEL, VELOC, DWELL, PMOVE, and CMOVE motion commands. Be aware that range limits still apply and errors may occur if a parameter contains a value out of range. The last few parameters are special purpose parameters. The DSM302 can load data into these parameters that might overwrite user data. The following table describes the function of the special purpose parameters.

Table 7-1. Special Purpose Parameters

Parameter Number	Special Purpose Function	Axis	Units
216-223	*Reserved*		
224	*Position Strobe 1*	Servo 1	user units
225	*Position Strobe 2*	Servo 1	user units
226	*Commanded Position* at Follower Enable Trigger	Servo 1	
227-231	*Reserved*		
232	*Position Strobe 1*	Servo 2	user units
233	*Position Strobe 2*	Servo 2	user units
234	*Commanded Position* at Follower Enable Trigger	Servo 2	
235-239	*Reserved*		
240	*Position Strobe 1*	Aux 3	user units
241	*Position Strobe 2*	Aux 3	user units
242-247	*Reserved*		
248-255	*Reserved*		

Parameters are all reset to zero after a power cycle or after a DSM302 configuration is stored by the PLC. Parameters can be assigned in three ways:

- The motion program LOAD command.
- The *Load Parameter Immediate* %AQ command.
- The COMM_REQ function block. This is the preferred way if you need to send multiple parameters per scan. The COMM_REQ function block is described in Appendix B.

Assigning a value to a parameter overwrites any previous value. Parameter values can be changed during program execution, but the change must occur before the DSM302 begins executing the Type 3 Cmd (Move, Wait or Dwell) previous to the Type 3 Cmd that uses the parameter. This is due to the pre-processing of program blocks that occurs within the DSM302. Note that a JUMP command clears pre-processing and forces a program block to be re-processed.

Below is an example of a motion program using Parameters. The values of Parameters 1-5 are pre-loaded with a COMREQ command from the PLC at least two program blocks before usage. (Remember that "program blocks" are not the same as sections of the motion program that are labeled with the BLOCK # command.)

Command	Data	Comments
BLOCK	1	
VELOC-P	1	Set velocity of first move = value in Parameter 1
ACCEL-P	2	Set acceleration of first move = value in Parameter 2
CMOVE-AL-P	3, ABS, LINEAR	Reach velocity of second move (20000) at position = Par. 3
BLOCK	2	
VELOC	20000	Set velocity of second move = 20000
PMOVE-IL	20000, INC, LINEAR	Normal PMOVE
DWELL-P	4	Dwell for Parameter 4 time
PMOVE-IL-P	5, INC, LINEAR	PMOVE to value in Parameter 5
(Strobe #1 occurs on Axis-1 during move to Param. 5 position)		
DWELL	1000	Dwell for one second
LOAD-P	6	Load Parameter 6 parameter
BLOCK	3	
PMOVE-IL-P	224, INC, LINEAR	Move to strobed position for Strobe #1 on axis-1
DWELL	2000	Dwell for two seconds
PMOVE-AS-P	6, ABS, SCURVE	Final stop position = value in Parameter 6

Calculating Acceleration, Velocity and Position Values

One method of determining the value for APM or DSM motion program variables such as *Acceleration*, *Velocity* or *Position* is to plot the desired move or move segment as a velocity profile. A velocity profile plots time on the horizontal axis of a graph and velocity on the vertical axis. The key to understanding profile generation is to break the complete move into smaller segments that may be analyzed geometrically. Most applications will use the economical trapezoidal move, velocity profile as illustrated below. To move as quickly as possible, use a triangular velocity profile if the servo has sufficient speed range. A triangular move would accelerate half the distance then decelerate the remaining half. Another alternative is to use a trapezoidal profile with a shorter slew segment.

Kinematic Equations

Kinematics is the branch of mechanics that studies the motion of a body or a system of bodies without consideration given to its mass or the forces acting on it. The following table includes transformations of the basic linear equations as applied to the acceleration portion of motion profiles. Use these formulae to calculate the velocity and acceleration for the acceleration portions of the move.

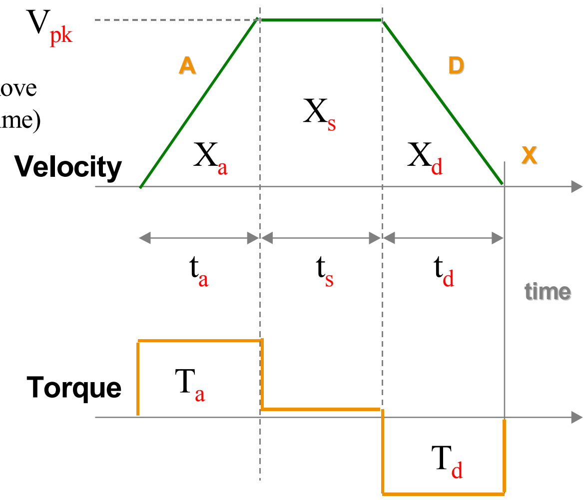
Table 7-2. Linear Equation Transformations

Given / *Solve For*	**A, X**	**A, V**	**A, t**	**V, t**	**V, X**	**X, t**
Acceleration				V/t	$V^2/2X$	$2X/t^2$
Velocity	$\sqrt{2AX}$		At			2X/t
X *(Distance)*		$V^2/2A$	$At^2/2$	Vt/2		
time	$\sqrt{2X/A}$	V/A			2X/V	

Let's take a look at the figure below. Beginning at zero velocity the axis will accelerate in a positive direction (t_a), run (slew) at velocity for some time (t_s), then decelerate back to zero velocity (t_d). That's a complete move or move segment. Looking at the figure below we can easily separate the different portions of the move. ***A common rule of thumb is to divide the trapezoidal move into three time portions, one-third for acceleration, one-third at slew velocity and the remaining third to decelerate.*** The slew (X_s) section of an equally divided trapezoidal velocity profile represents ½ of the distance moved and the acceleration and deceleration portions each represent ¼ of the total distance. The rule of thirds minimizes the RMS torque current in the motor and is the most economical use of energy.

Trapezoidal Move

- Limits max motor speed
- Higher accel torque than triangle move
- Symmetrical profile (1/3, 1/3, 1/3 time) maximizes power transfer to load
- Most common for long moves



A = acceleration
D = deceleration
X = distance
V_{pk} = velocity peak
t_a = time acceleration
t_s = time at slew velocity
t_d = time deceleration
T_a = acceleration Torque
T_d = deceleration Torque
X_a = acceleration distance
X_s = slew distance
X_d = deceleration distance

Equations

$$x = V_{pk}\left(0.5t_a + t_s + 0.5t_d\right)$$

$$V_{pk} = \frac{x}{\left(0.5t_a + t_s + 0.5t_d\right)}$$

$$a = \frac{V_{pk}}{t_a} \Rightarrow D = \frac{V_{pk}}{t_d}$$

Figure 7-22. Trapezoidal Move

Once the move segment outline is drawn, we need to examine specifications or physical restrictions applicable to the move. For instance the move may have to complete in a certain time interval (t_a + t_s + t_d) or move a fixed distance (X). The maximum velocity (V_{pk}) of the servomotor is one example of a physical limitation. Given any two known values of the acceleration portion of the move segment, a remaining variable can be found using the kinematic equations as illustrated in the example below.

Trapezoidal Velocity Profile Application Example

Let's assume that a complete move of 16 inches must be made in three seconds and the maximum motor velocity, translated through the gearing is 15 inches per second. Using our rule of thumb, we divide the move's time into thirds: t_a = 1sec, t_s = 1sec and t_d = 1sec. We can also subdivide the 16 inch move into three distances. The slew (X_s) section of an equally divided trapezoidal velocity profile represents ½ of the distance moved and the acceleration (X_a) and deceleration (X_d) portions each represent ¼ of the total distance: X_a =4 in, X_s =8 in and X_d =4 in.

To calculate peak Velocity (V_{pk}), the first acceleration portion of the move must travel a given distance (X_a) in a given time (t_a). From the above Kinematic Velocity formula (2X/t) using the given, X_a =4 inches and t_a = 1 second, (2*4 inches) / 1 second = 8 inches/second.

To calculate Acceleration the simplest formula is (V/T)=(8 inches/second / 1 second)=8 inches/second/second.

The Position (Distance = X) is the entire distance moved (X_a + X_s + X_d) or 16 inches

Triangular Velocity Profiles

The triangular velocity profile minimizes servo acceleration rate and requires a higher servomotor velocity when compared to a trapezoidal profile of the same distance and time. Use a triangular profile for fast short moves.

Equations:

$$x = \frac{1}{2} V_{pk} (t_a + t_d)$$

$$V_{pk} = \frac{2(x)}{(t_a + t_d)}$$

$$a = \frac{V_{pk}}{t_a}$$

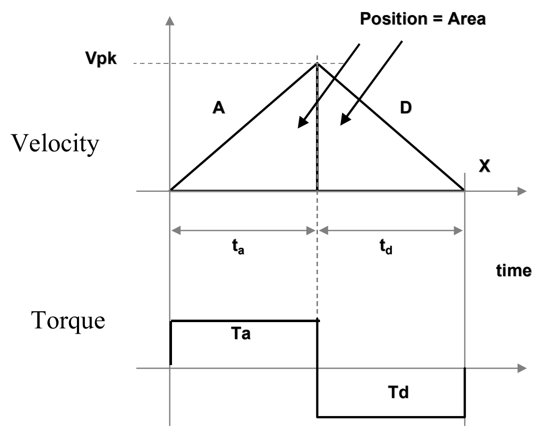


Figure 7-23. Triangular Velocity Profile

Non-Linear or S-Curve Acceleration

S-Curve or jerk limited acceleration calculation is simple to do if the linear calculation is accomplished first. The APM and DSM motion controllers use 100% jerk limiting. To convert a linear acceleration to 100% jerk limited acceleration you either double the Acceleration value (2*A) or double the time used for acceleration ($2t_a$). Using S-Curve acceleration at the same acceleration rate (A) as linear acceleration will require twice the time (t_a) reaching velocity. If the time duration of the move must remain the same and the servo has sufficient peak torque, use twice the acceleration (2*A) to reach velocity in the same amount of time.

Equations:

$$X_a = X_d = \frac{V_{pk}^{\ 2}}{a}$$

$$t_a = t_d = \frac{2V_{pk}}{a}$$

$$t_s = \frac{X_s}{V_{pk}}$$

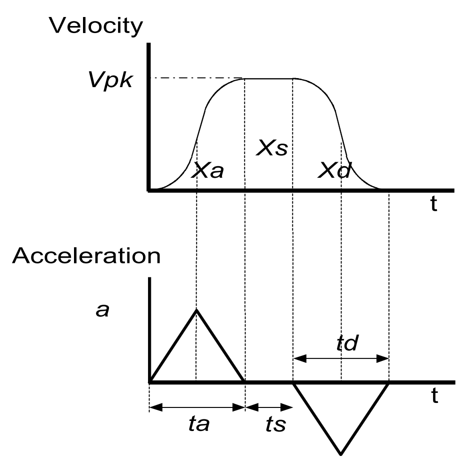


Figure 7-24. S-Curve Acceleration

Chapter 8 Follower Motion

Configuring the DSM302 for ***Control Loop*** = FOLLOWER allows each Servo Axis (slave) to respond to a Master Axis input using a programmable slave **:** master ratio. The DSM302 defines the slave **:** master ratio as the ratio of two integer numbers A and B. The basic formula for computing Follower motion is:

Follower Servo Axis motion (slave axis) = Master Axis motion x (A/B)

or

slave : master ratio = A : B ratio

If a *Jog*, *Move at Velocity* or *Execute Motion Program* command is also initiated, the axis motion will represent the combination of the Master Axis motion and the internally commanded motion. This Chapter provides details of servo motion related to the Master Axis input. Refer to Chapter 9 for additional information about combined Follower and commanded motion.

When the *Enable Follower* %Q bit is turned ON, an axis will immediately begin following the selected ***Master Source*** unless an external ***Enable Trigger*** input has been selected. If an external ***Enable Trigger*** input has been selected then the *Enable Follower* %Q bit must be ON and an OFF to ON transition of the trigger input must occur. The external trigger input CTL01 - CTL016 is selected in the configuration software.

There are two ways to provide acceleration control when following a master source.

1. External means, such as ramping the velocity of the master input, can be used to limit acceleration.
2. Select ***Follower Mode*** = RAMP in the configuration software to make the Follower axis ramp up at the ***Jog Acceleration*** rate.

Master Sources

A DSM302 Servo Axis can follow one of several master input sources:

- The external Aux Axis 3 encoder input
- The internal master velocity generator
- The Aux Axis 3 analog input 1
- Servo Axis 1 can select the Servo Axis 2 encoder to be its master input source

If **ENC3/INT** is selected in the configuration software for an axis, the *Select Internal Master* %Q bit is used to choose between the external Aux Axis 3 encoder input and the internal master generator.

Note that follower motion is summed with *Jog*, *Move at Velocity,* or Motion Programs. If an axis is following the internal master at velocity V1, and a *Jog* is commanded at velocity V2, the axis will move at velocity V1 + V2.

Aux Axis 3 (ENC 3) Master Input

When **ENC3/INT** is configured as the ***Master Source*** and the *Select Internal Master* %Q bit is OFF, the Aux Axis 3 encoder will be the master input source.

The DSM302 always tracks and reports the position of the encoder in *Actual Position* (Aux Axis 3) %AI data. The DSM302 also computes and reports the encoder velocity in *Actual Velocity* (Aux Axis 3) %AI data.

The Aux Axis 3 Encoder has configured ***High and Low Count Limits***. When a limit is reached, the position rolls over to the other limit and continues changing.

Performing a Home Cycle for the master encoder can initialize actual Position for Aux Axis 3. **An Aux Axis 3 encoder Home Cycle does not generate any motion**. Instead, it sets the *Actual Position* %AI data to the configured master source ***Home Position*** at the next marker pulse when the Home Switch input is ON. It also sets the *Position Valid* (Aux Axis 3) %I bit when it sets the position. See Chapter 5, %Q Discrete Commands, *Find Home* (Aux Axis 3) for details about the Aux Axis 3 encoder Home Cycle.

Example 1: Following Aux Axis 3 Encoder Master Input

In this example, a graph of velocity (v) versus time (t) shows the velocities of the master input (Aux Axis 3), and the axis that is following the master. The DSM302 is configured with ***Master Source*** **ENC3/INT** and the *Select Internal Master* %Q bit is OFF. The A:B ratio is 1:1. The velocity profile of the following (slave) axis is identical to the master input.

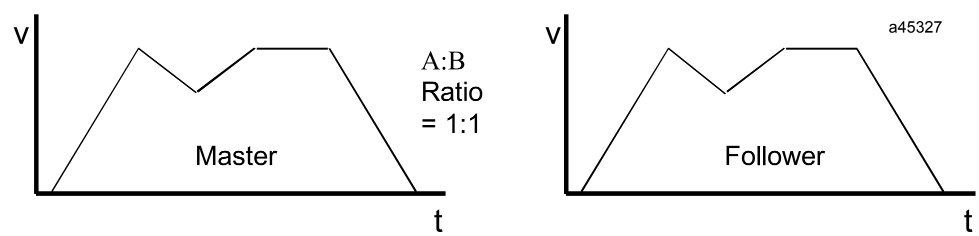


Figure 8-1. Following Encoder 3 Master Input

Internal Master Velocity Generator

The DSM302 can internally generate a velocity up to one million counts per second. When **ENC3/INT** is configured as the ***Master Source*** and the *Select Internal Master* %Q bit is ON, the internal master will be used as the master input source. The velocity generated is initially zeroed. The *Internal Master Velocity* %AQ command can be used to set the generated velocity.

There is no acceleration control of the internal velocity generator. One method of preventing the instantaneous acceleration attempt when *Enable Follower* is turned ON is to incrementally step up the internal velocity using the *Internal Master Velocity* %AQ command.

Example 2: Following the Internal Master

When following the internal master, the following axis simply moves at the current internal velocity. In this example, dotted lines indicate the times when a master velocity change takes place using the *Internal Master Velocity* %AQ command. The DSM302 is configured with ***Master Source*** **ENC3/INT** and the *Select Internal Master* %Q bit is ON. The A:B ratio is 1:1.

The velocities commanded are the following: initially 0, then 11 thousand, 15 thousand, 4 thousand, 7 thousand, and finally 0 again. The velocity of the following axis is identical to the internal master velocity.

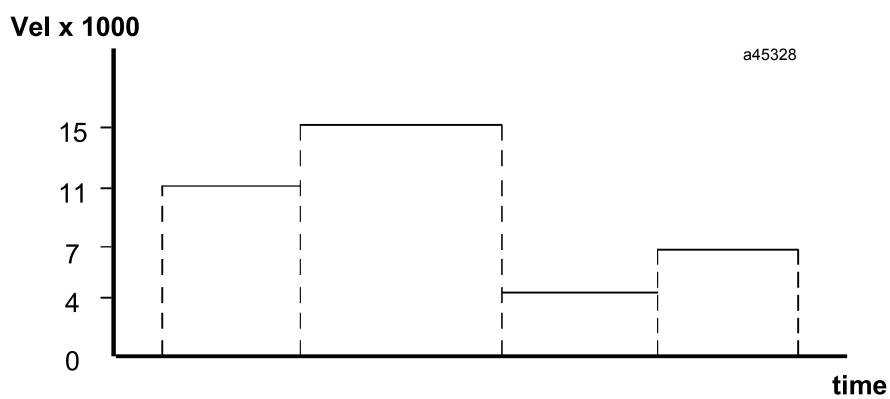


Figure 8-2. Following the Internal Master

Analog Input Master

When **ANALOG** is configured as the ***Master Source***, the faceplate Aux Axis 3 *Analog Input 1* is the master input source. The DSM302 converts the *Analog Input* into a velocity command. The ***Analog Maximum Velocity*** configuration parameter sets the velocity generated when the *Analog Input* is +10V. Zero velocity is generated when the *Analog Input* is 0V, and the negative of the maximum velocity is generated at –10V.

Example 3: Following the Analog Input

In the graph below, the DSM302 is configured with *Master Source* **ANALOG**. The A:B ratio is 1:1. The dotted lines indicate when the *Enable Follower* %Q bit was turned ON and OFF.

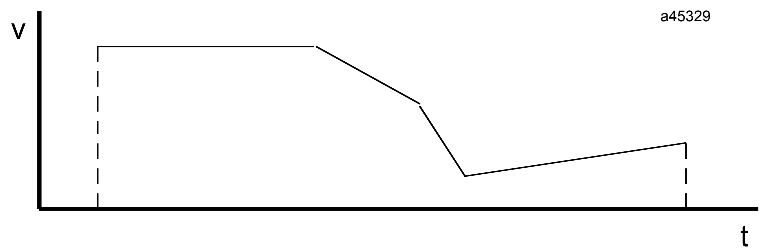


Figure 8-3. Following the Analog Input

Encoder (Servo Axis 2) Master

The DSM302 can be configured so that Servo Axis 1 will follow Servo Axis 2. With “ENC2” as the ***Master Source*** for axis 1, axis 1 will follow axis 2 at the axis 2 velocity multiplied by the A:B ratio configured for axis 1.

Example 4: Following Servo Axis 2 Encoder

Axis 1 of a DSM302 is configured with *Master Source* “ENC2”. With the A:B ratio 1:2, axis 2 is commanded to *Move at Velocity* 12000 and then 0. Axis 1 follows axis 2 at half of the axis 2 velocity and acceleration, and moves only half the distance that axis 2 has moved.

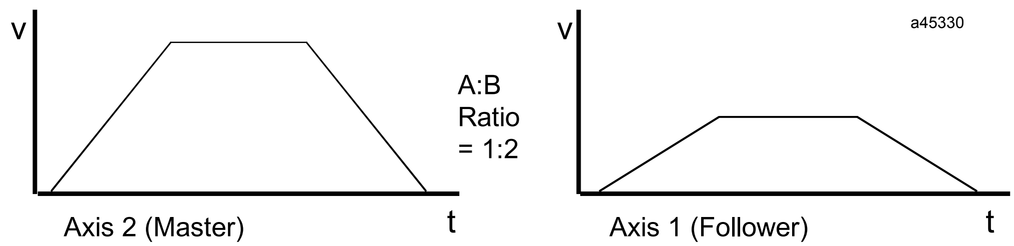


Figure 8-4. Following Servo Axis 2 Encoder

A:B Ratio

A DSM302 axis following a master input can do so at a wide range of slave **:** master (A:B) ratios. The “A” value can be any number from –32768 to 32767. The “B” value can be anywhere between 1 and 32767. The magnitude of the A:B ratio can be from 1:10,000 to 32:1. Thus very precise ratios such as 12,356:12,354 or 32,000:1024 can be used.

The *Follower A/B Ratio* %AQ command can be used to change the A:B ratio at any time, even while following. However, an invalid ratio will generate a status error and be ignored. An invalid ratio is a ratio with B equal to or less than zero or A:B magnitude greater than 32:1 or less than 1:10,000.

Ratios from 1:1024 to 1024:1 can be obtained with a DSM302 if the slave axes are cascaded. When axis 2 follows encoder 3 and axis 1 follows axis 2, if both ratios are 32:1 the slave:master ratio of axis 1 to encoder 3 is 1024:1.

When following with a non 1:1 ratio, the velocity profile of the master and follower will look somewhat different. Horizontal lines, indicating constant velocity, and slanted lines, indicating acceleration and deceleration, will be different. **If the A:B ratio is less than 1:1, the follower velocity and acceleration will be less than the master. Likewise, if the A:B ratio is greater than 1:1, the follower velocity and acceleration will be greater than the master.** The duration of motion, and time that the slave axis will accelerate, decelerate, or stay at constant velocity are the same for the master and follower.

The distance moved, which in a velocity profile is the area between the graph and the time axis, will be that of the master multiplied by the A:B ratio. If A is zero, no following motion will be generated. If A is negative, the following axis will move with the direction of motion reversed.

Example 5: Sample A:B Ratios

All of the following samples are following the master source input at various A:B ratios.

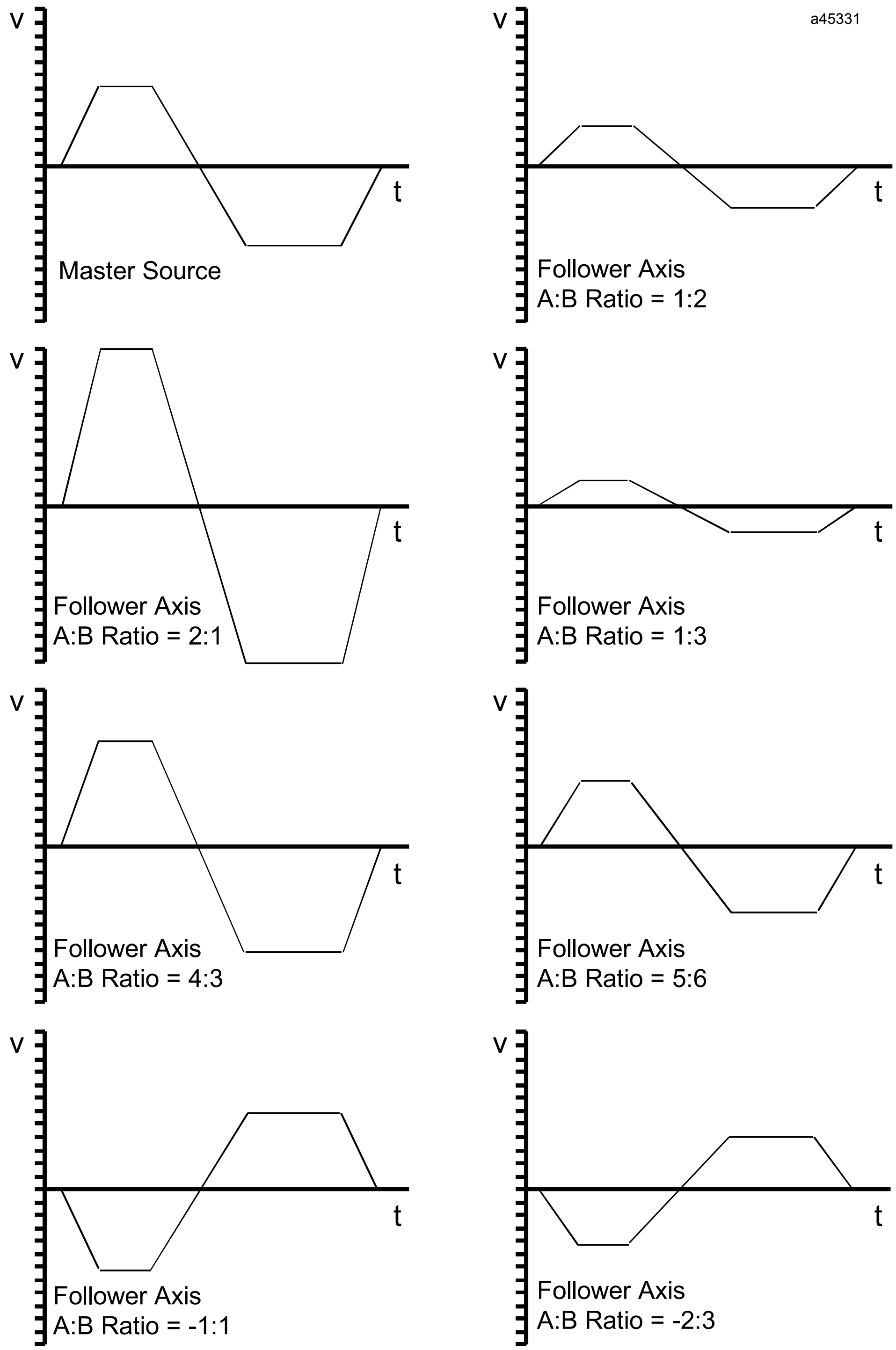


Figure 8-5. Sample A:B Ratios

Example 6: Changing the A:B Ratio

One example of variable A:B ratios is to use one ratio while moving positive, and another when moving negative. **Note that determination of positive and negative velocity and update of the A:B ratio must be done in the PLC**. In the profile below, the following axis uses a 2:1 ratio when moving positive and a 1:2 ratio when moving negative.

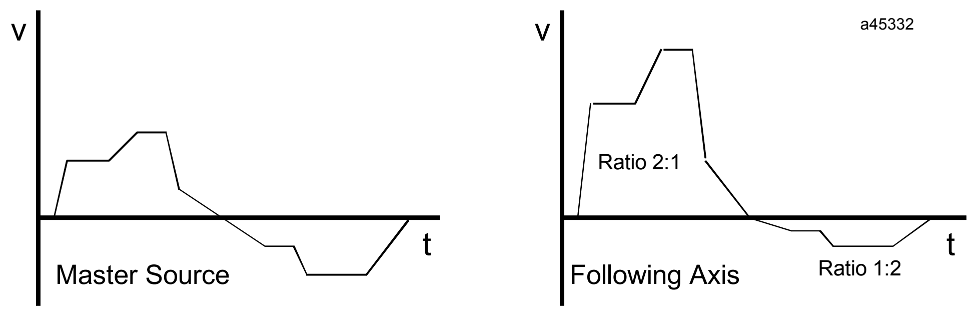


Figure 8-6. Changing the A:B Ratio

Example 7: Ratio Stepping

Another example of variable A:B ratios is step up the ratio as a type of acceleration control. Initially setting a ratio 1:32, and incrementing the numerator after a delay would step the following axis from 1:32 to 2:32 to 3:32 and so forth all the way to 32:32 which is 1:1. In this example, the PLC increments the A:B ratio by 1/16th every second until it reaches 1:1. The dotted lines represent ratio changes, the dashed line indicates when the *Follower Enable* %Q bit is turned OFF.

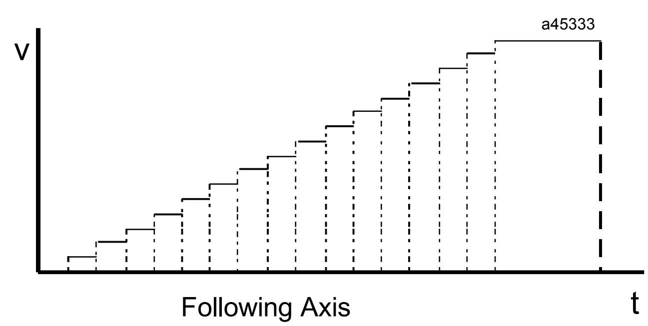


Figure 8-7. Ratio Stepping

Velocity Clamping

Velocity clamping is available using the ***Velocity Limits*** set in the Configuration software. **When the master velocity exceeds the configured limit, the following axis will continue to move at the limit velocity multiplied by the A:B ratio.** The *In Velocity Limit* %I bit is set and a status error is generated to indicate that the slave axis is no longer locked to the master input positioning. The slave axis has essentially *fallen behind* the master input. The ***Velocity Limit*** units are counts per millisecond. Thus a limit of 400 cts/ms will limit velocity to 400,000 cts/sec.

Example 8: Velocity Clamping

The ***Velocity Limits*** are set to 100 and –100 in this example. Thus the master input velocity is clamped at 100,000 cts/sec in either direction. When the master axis peaks greater than the limits, the following axis stays at the limit. After the master slows to under the limit, the following axis continues tracking the master axis velocity. Counts generated in excess of the ***Velocity Limits*** are lost to the follower. The horizontal dashed lines indicate the velocity limits. The shaded area indicates the times when the *In Velocity Limit* bit is ON and the following axis is falling behind the master.

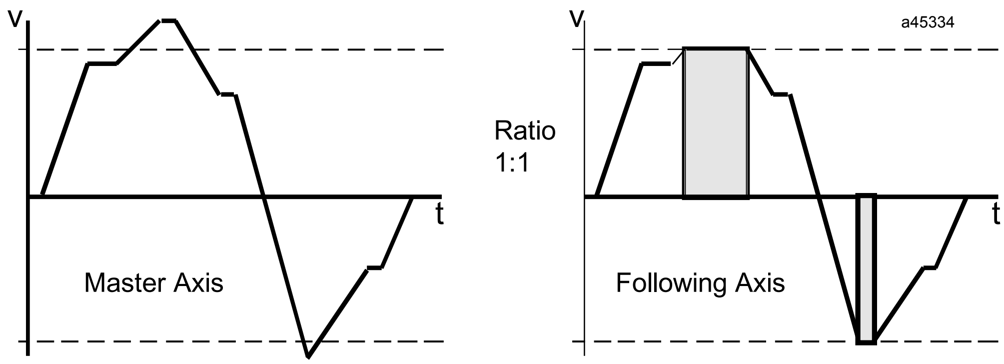


Figure 8-8. Velocity Clamping

Unidirectional Operation

Setting one of the ***Velocity Limits*** to zero results in unidirectional follower motion. Any master axis counts in the zero limited direction are ignored. When a ***Velocity Limit*** is zero, no error is generated by counts in the zero limited direction. The *In Velocity Limit* %I bit, however, does reflect the presence of counts in the zero limited direction.

Example 9: Unidirectional Operation

In this example, the negative ***Velocity Limit*** is set to zero. As shown in the velocity profile below, the following axis follows the positive counts, but ignores the negative counts. Note that when the master is moving negative, the *In Velocity Limit* %I bit is ON, but no status error is generated.

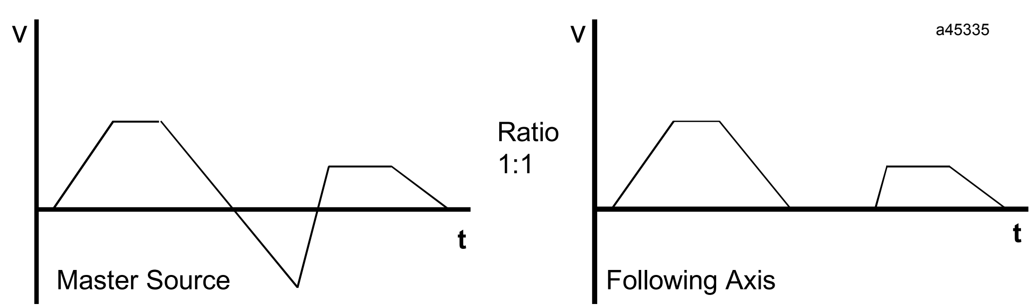


Figure 8-9. Unidirectional Operation

Synchronizing Aux Axis 3 Velocity and the Internal Master

Synchronization of the external Aux Axis 3 encoder velocity and the internal master velocity may be needed to prevent the follower control loop from commanding large accelerations when the *Select Internal Master* %Q bit is switched OFF. This synchronization can be achieved by using the PLC to monitor the difference between the commanded internal master velocity and the reported Aux Axis 3 *Actual Velocity*. The PLC can ramp the *Internal Master Velocity* %AQ command in small steps until the internal master velocity and the Aux Axis 3 *Actual Velocity* are close to each other. At that point the PLC can turn off the *Select Internal Master* %Q bit to switch the follower loop master input to the external Aux Axis 3 Encoder.

Example 10: Aux Axis 3 Velocity and Internal Master Synchronization

To synchronize the internal master and Aux Axis 3 velocity, the *Internal Master Velocity* is stepped up to an expected Aux Axis 3 encoder velocity. When Aux Axis 3 *Actual Velocity* is within 128 cts/sec of this velocity, the PLC program will turn the *Select Internal Master* %Q bit OFF to switch the master source to the Aux Axis 3 encoder. The shaded area indicates the time when the *Select Internal Master* %Q bit is ON.

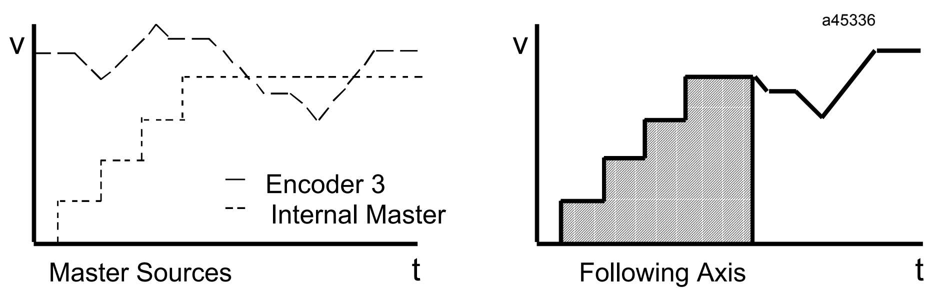


Figure 8-10. Encoder 3 and Internal Master Synchronization

Enabling the Follower with External Input

An external trigger signal applied to one of the inputs CTL01 - CTL16 can be used to enable the follower axis. The input used for the Enable Trigger is selected with the configuration software. When no input is selected, the follower is enabled and disabled directly by the *Enable Follower %Q* bit. When an input is selected for the Enable Trigger and the *Enable Follower* %Q bit is set, the next positive transition of the defined input will instantly enable the follower. The follower will remain enabled until the *Enable Follower* %Q bit is cleared. The faceplate 24v inputs have 5 ms filters that result in a Follower Enable Trigger response time of 5-7 milliseconds. The faceplate 5v inputs do not have these filters and will provide an Enable Trigger response time of 1 millisecond or less.

When the *Enable Follower* trigger occurs, the *Commanded Position* at that point is captured in a parameter register so that it can be used in a *Programmed Move* command. The position is captured in parameter 226 (for Servo Axis 1) or parameter 234 (for Servo Axis 2).

Follower Enabled status is returned in %I bit offset 28 (Servo Axis 1) or %I bit offset 44 (Servo Axis 2).

Follower Axis Acceleration Ramp Control

For applications where the Follower is enabled when the Master command is already up to speed, the Follower Ramp feature can be used to apply a controlled acceleration rate to bring the follower axis up to speed. This may be done without losing any Master command counts from the point at which the Follower was enabled. During the automatically generated Follower Ramp Control make-up move, the acceleration/deceleration does not exceed the active jog acceleration value and provides a smooth motion. When the Follower Ramp feature has been selected (FOLLWR MODE is set to ACC RAMP) and the follower is enabled, the following axis is ramped up to the master velocity at the active ***Jog Acceleration*** rate. This function is most useful when the master source is in motion before the follower mode is enabled. In addition to the PLC Enable Follower %Qbit, a PLC initiated control bit (CTL9-CTL12) or a high-speed DSM302 module input (CTL01-CTL08, CTL13-CTL16) may be configured as the enable follower signal for position registration functions. When the *Enable Follower* %Qbit is ON, then the CTL bit chosen acts as a rising edge trigger to enable follower mode. After Follower is enabled, only the PLC *Enable Follower* %Qbit controls the active state of the following function. When the follower axis is enabled to a moving master source, some master source counts cannot be used immediately. The master counts that accumulate during acceleration of the follower axis are stored. When the follower axis reaches the master velocity, they will be inserted during make-up distance correction motion. This motion has an automatically calculated trapezoidal velocity profile determined by the ***Follower Ramp Distance Makeup Time***, the amount of accumulated counts, and the active jog acceleration at the beginning of the correction. Set the Follower ***Follower Ramp Distance Make-up Time*** to the desired time in the configuration software or it can be changed with the PLC %AQ Command 42h.

If the ***Follower Ramp Distance Makeup Time*** is too short then the automatically generated velocity profile is triangular in profile. If during the distance correction velocity exceeds 80% of the velocity limit, then the automatically calculated move velocity will be clamped at 80% of the configured velocity limit. Clamping the makeup move velocity at 80% of the velocity limit allows the system some reserve velocity capacity for continued tracking of the master source velocity. In both cases a warning message is reported and the real distance make-up time is longer than programmed, but the distance is still corrected properly.

Setting a ***Follower Ramp Distance Make-Up Time*** of 0 allows the Ramp feature to accelerate the axis without making up any of the accumulated counts. In this instance velocity will not exceed the master velocity. For applications where lost counts do not matter, set the distance make-up time = 0.

By default the superimposed motion profile that is automatically generated by the follower ramp function (with non-zero makeup time) is trapezoidal using the active jog acceleration and a distance derived from the active Acceleration Ramp Makeup Time. If trapezoidal correction is performed, the effective make-up time increments by 8ms and can have a minimum value of 40 ms.

The value of the Master Velocity Limit (+Vlim or –Vlim) may affect functionality differently depending on the relationships of the master source (Enc3, Internal Master, Analog Master) velocity. The following case examples illustrate these points.

Case 1: The master source velocity is less than 80% of the configured Vlim and the makeup time (Mkup Time) is a long enough interval so that the resultant velocity remains less than 80% of the Vlim. This is the preferred operation, no errors are reported and the over speed move of the ramp function occurs within the specified makeup time. The follower axis velocity will not exceed 80% of the Vlim unless the master source velocity increases.

Case 2: The master source velocity is below 80% of the configured Vlim but the makeup time interval is too short to allow operation as in case 1. A status only error (ECh) will be returned when the follower velocity matches the master command velocity. The makeup move will accelerate using the active Jog acceleration to 80% of the velocity limit (Vlim). The makeup move will occur and all accumulated counts stored during initial acceleration will be used. The makeup move will be a multiple of 8msec longer in duration than the active makeup time.

Case 3: The master source velocity is greater than 80% of the configured velocity limit (Vlim) when the follower velocity matches the master command velocity. A status only error (EAh) is returned and no makeup correction move is attempted.

Case 4: At the time when the follower velocity matches the master command velocity and the makeup move is to occur and conditions are the same as in Case 1 or Case 2 and the makeup move has initiated, the master source increases to >80% of the velocity limit. The amount of accumulated counts and the active makeup time value will determine if the makeup move will complete in the specified makeup time. A status only error (F2h) will occur if the combined master command velocity and the makeup move velocity reach 100% of the velocity limit. The master command velocity will not exceed 100% of the Vlim value. Accumulated counts may be lost and the makeup move will not complete.

When the Follower is enabled by the *Enable Follower* %Q bit and an optional Trigger input, the Follower Axis Velocity will be ramped up at the ***Jog Acceleration*** rate. When the *Enable Follower* %Q bit is turned off, the axis velocity will ramp down at the same rate. The *Moving* %I bit indication is turned on while the ramp control is in effect for both the ramp up/make-up and ramp down.

The *Follower Enabled* and *Moving* %I bits can be monitored by the PLC to determine which part of the follower ramp up/ramp down cycle is active. The following figure shows the state of *Follower Enabled* and *Moving* during a follower cycle.

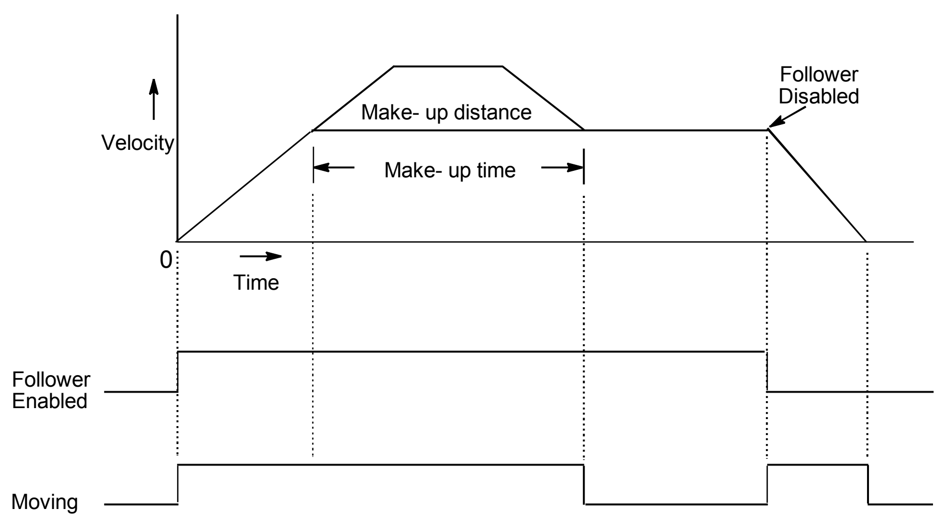


Figure 8-11. Follower Ramp Up/Ramp Down Cycle (Case 2)

The programmed make-up time can be too short for the required distance correction. In this case a warning error is reported (in the point B of the trajectory), but system continues acceleration up to the speed, insuring the minimum possible distance correction time . The velocity profile for such case is shown on the figure 8-12.

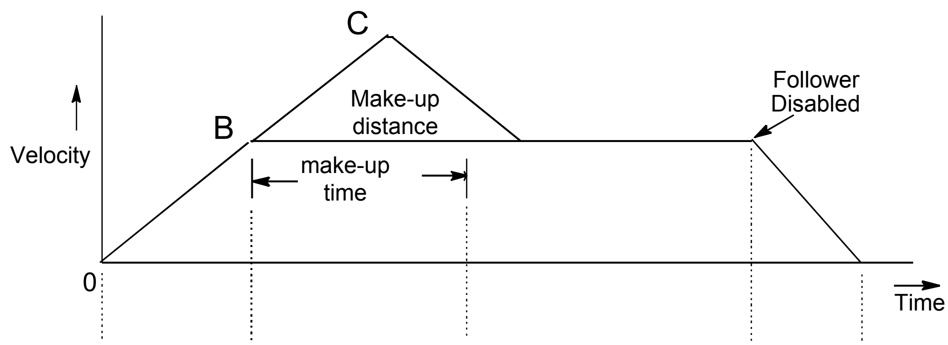


Figure 8-12. Follower Ramp Up/Ramp Down Cycle – Case 2 with make-up time too small.

During the ramp phase of the distance correction, the velocity limit is controlled. If calculated velocity is too high, then the velocity is clamped and warning error code is set (in the point C of the trajectory). Figure 8-13 shows the velocity profile during the follower ramp cycle for this case.

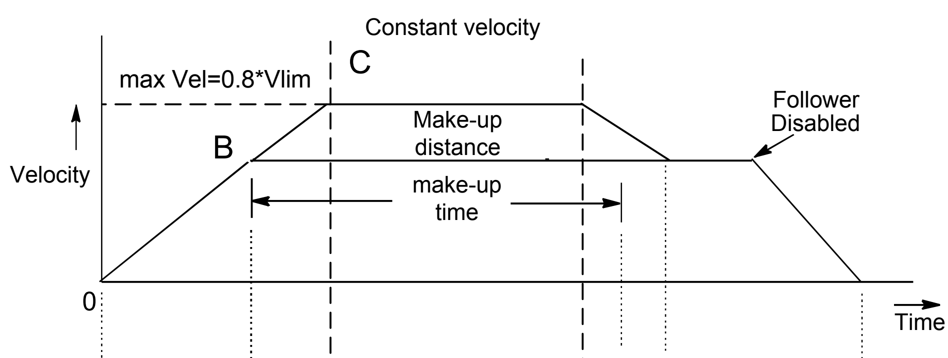


Figure 8-13. Follower Ramp Up/Ramp Down Cycle - case with active velocity limit.

If the acceleration time (sector BC of the trajectory in figure 8-13) exceeds 128 seconds, then another warning error will be reported. In this case the distance also will be corrected accurately.

Follower Winder

Introduction

The Follower Winder feature is a Traverse or Spool Winder and can be thought of as an “electronic fishing reel”. With this feature selected, the slave axis will follow within a specified distance zone and then at the end of the zone, reverse direction and traverse the distance in reverse. This cycle will repeat indefinitely. The follower axis will always stay within the zone for both directions of master commands. During the winding motion, the PLC can send zone adjustments to accommodate winder reel taper on either or both sides of the reel.

Configuration

Follower Winder Mode Enable: Use the configuration software to select ***Follower Mode*** = WINDER with the desired taper mode. If desired, use the *Select Return Data* %AQ command to select the Zone Position Register to be reported in the *User Selected Data* %AI location.

Note

Taper is only applied when the PLC sends a different winder zone length.

Winder Zone Length: Use the configuration software or *Follower Winder Zone Length* %AQ command to establish the initial move zone length for the winder motion. The *Follower Winder Zone Length* %AQ command may be sent later during the winding operation to alter the zone length to produce the desired taper.

Note

In WINDER Mode only, the following formula applies to determine a valid combination of the follower (B/A) ratio and the winder zone length:

Follower (B/A) Ratio * (Winder Zone Length) <= 1,073,741,823 counts

Operation Steps

1. With the *Enable Follower* %Q bit OFF, set the ***Winder Zone Length*** to the count value representing the width of the spool or the distance to be traversed in one direction for the winding operation.

Note

The following formula applies when issuing a "Winder Set Position" command:

Follower (B/A) Ratio * (Winder Set Position Value) <= 1,073,741,823 counts

2. Command the follower axis to the winding start point and then send the *Set Winder Position* %AQ immediate command to synchronize the axis position to the winder zone and define the starting position within the zone. The magnitude of this preset value indicates the axis position

orientation within the winder zone. Zone position = 0 corresponds to the extreme left (most negative) side of the zone and Zone position = + (zone length – 1) corresponds to the extreme right side of the zone. The sign of the preset value determines the initial direction of movement. More specifically, when positive master commands are followed, a positive position preset will cause the winder axis to first move in the positive direction and a negative position preset will cause the winder axis to start in the negative direction. The reverse of this is true when the master command is negative.

3. Set the *Enable Follower* %Q bit (and the optional Enable Trigger bit if configured) to start the axis winder motion in response to master commands.

Operation Description

Once the follower is enabled, the winder axis will follow the master command to one end of the move zone and then reverse and proceed to the other end for as long as the master commands are in the same direction. If the master command reverses direction, the winder axis will follow in the reverse direction until the end of the zone is encountered where it will reverse direction and continue moving within the zone. The Zone Position Register always indicates the relative position of the winder axis within the zone. If its sign is +, then it is moving in the same direction as the master command and if its sign is **–**, it is moving in the opposite direction of the master command. For zone tapering, the PLC must send the new zone width at the correct time to produce the desired taper contour. If the taper mode selection = **left** or **right** side, the total zone change adjustment will be applied to the designated side of the zone. If the taper mode selection = **both** sides, the adjustment will be applied equally to each end of the winder zone.

During the winding operation, the PLC may send the ***Winder Zone Length*** %AQ command to adjust for winder spool taper. If the change in the zone length (absolute value) is greater than the distance from the winder axis position to the end of the zone, the DSM302 will wait to perform the update until the axis has moved to an acceptable position. Successive zone length change commands before the previous one has been updated will result in a reported error. A command for a zone length change > 25% while the follower is enabled will also result in a reported error.

Changing A/B Ratio in Winder Mode

In WINDER mode the A/B ratio cannot be changed unless the follower is disabled. Any attempt to change it when the follower is enabled will result in a reported error. The sign of A is determined by the sign of the winder zone position register, therefore, the user programmed sign of A is ignored in winder mode. The Zone Position Register represents the master command zone required to produce the configured winder axis zone, therefore, when the A/B ratio value is changed, the size of the Zone Position Register is adjusted proportionately. The zone move distance of the winder axis always remains equal to the configured zone length and also any preset position within the zone remains the same.

Follower Mode Master Axis and Connection Options

The diagrams on the following pages illustrate a variety of Master axis and Follower slave axis loop connection options.

The diagram below illustrates the two axes of the DSM302 connected in cascade with encoder 3 or the internal master as the master source for axis 2, and encoder 2 the master source for axis 1.

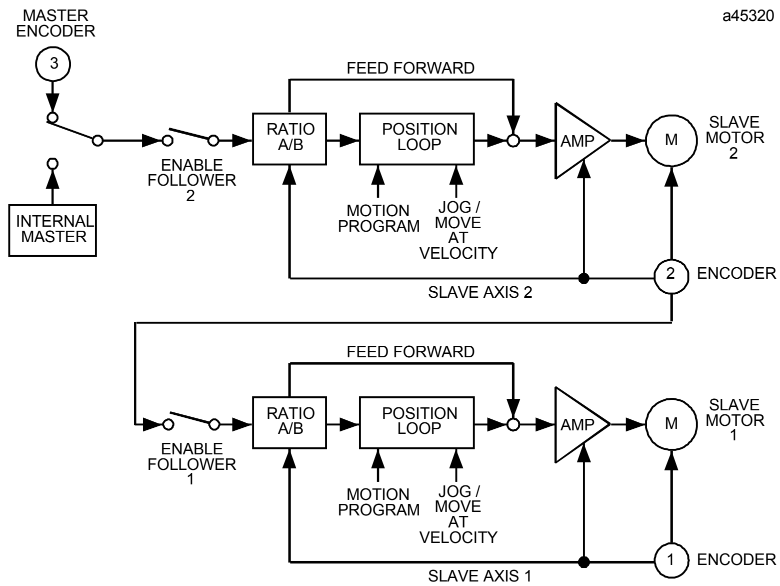


Figure 8-14. 2-Axis Cascade, Master Source Encoder 3/Internal Master

The diagram below illustrates the two axes of the DSM302 connected in parallel with encoder 3 or the internal master selectable as the master source for axis 1 or axis 2.

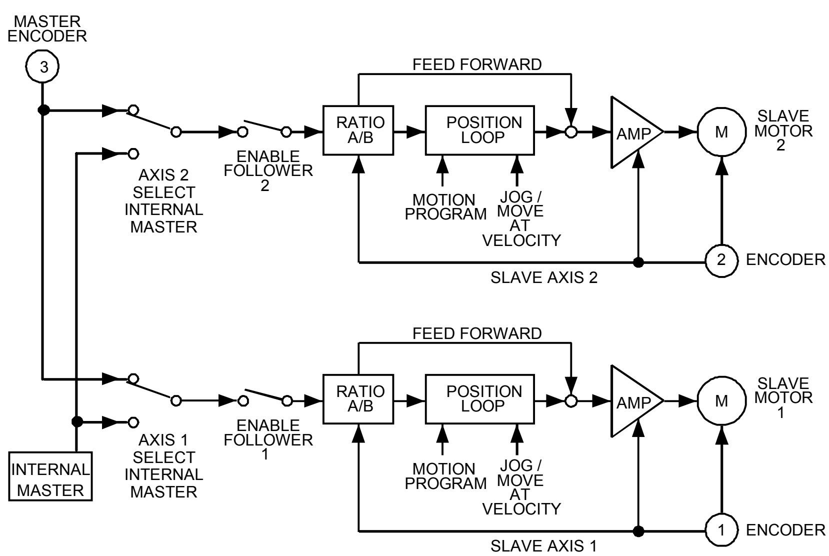


Figure 8-15. 2-Axis Parallel, Master Source Encoder 3/Internal Master

The following diagram illustrates the two axes of the DSM302 connected in cascade with the analog input as the master source for axis 2, and encoder 2 the master source for axis 1.

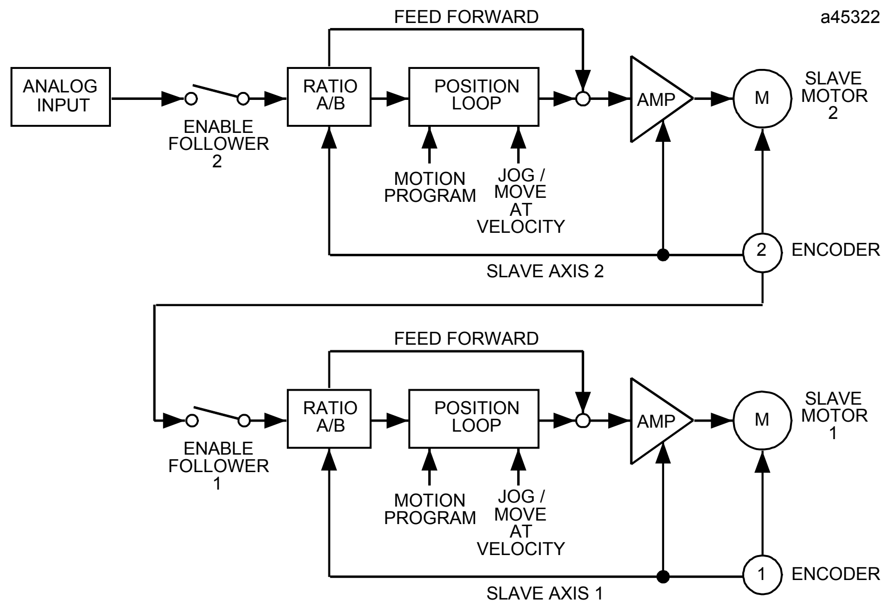


Figure 8-16. 2-Axis Cascade, Master Source Analog Input

The following diagram illustrates the two axes of the DSM302 connected in parallel with the analog input as the master source for both axis 1 and axis 2.

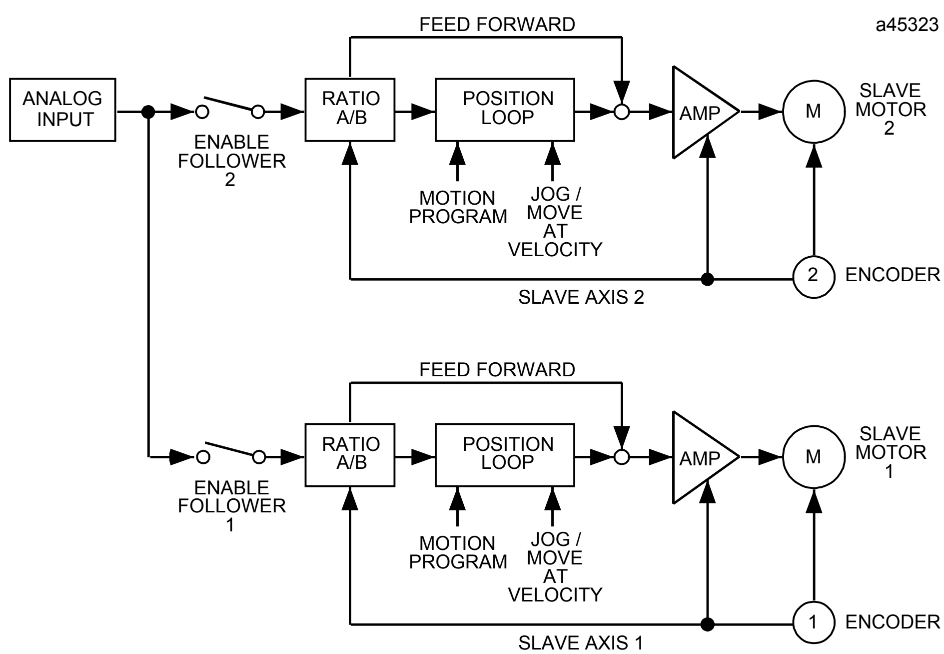


Figure 8-17. 2-Axis Parallel, Master Source Analog Input

The diagram below illustrates the two axes of the DSM302 connected with encoder 3 or the internal master as the master source for axis 2, and the analog input the master source for axis 1.

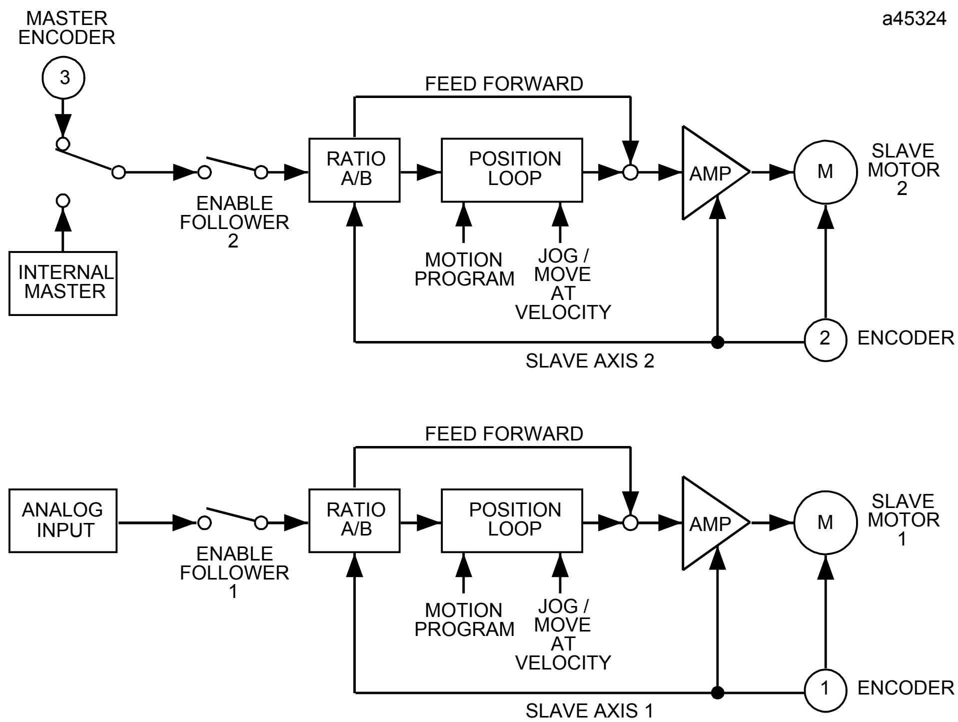


Figure 8-18. Axis 2 Master Source Encoder 3/Internal Master, Axis 1 Master Source Analog Input

The diagram below illustrates the two axes of the DSM302 connected with the analog input the master source for axis 2 and encoder 3 or the internal master as the master source for axis 1.

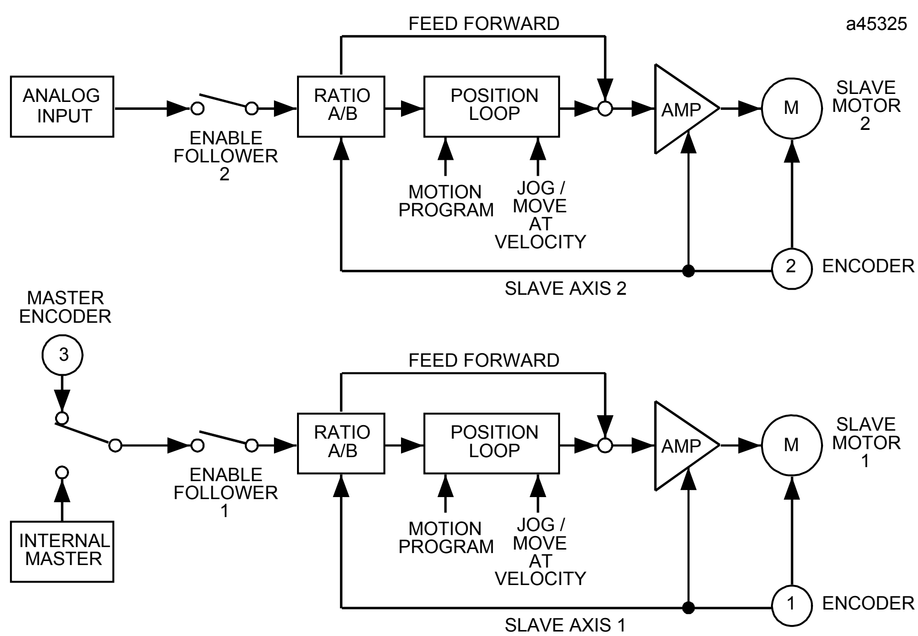


Figure 8-19. Axis 2 Master Source Analog Input, Axis 1 Master Source Encoder 3/Internal Master

Follower Control Loop Block Diagram

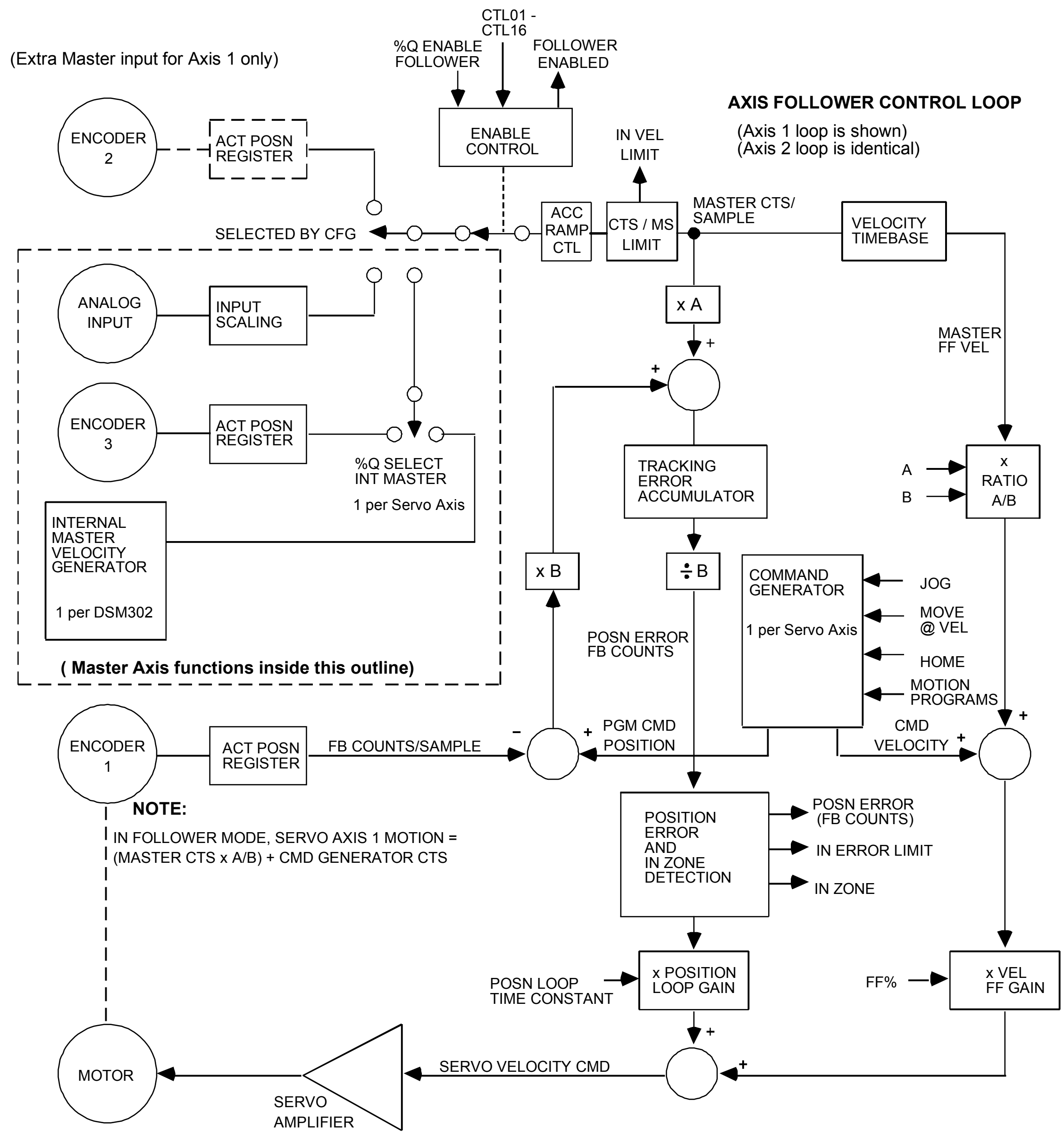


Figure 8-20. Axis Follower Control Loop Block Diagram

Chapter 9 Combined Follower and Commanded Motion

Combined motion consists of Follower motion commanded from a master axis combined with one of these internally commanded motions:

- *Jog Plus/Minus* %Q Command
- *Move at Velocity* %AQ Command
- *Move* %AQ Command
- Stored Motion Program

Combined motions are additive. The slave axis motion is equal to the sum of the motion commanded by the master axis and the internally commanded motion.

Example 1: Follower Motion Combined with Jog

In this example, the *Enable Follower* %Q bit is set, causing the slave axis to follow the master input. While the slave axis is following, the *Jog Plus* %Q bit is set. The following axis accelerates from its master's velocity to its master's velocity added to the current ***Jog Velocity***. This acceleration will be just as if the axis was not following a master source at the time. When the *Jog Plus* %Q bit is cleared, the following axis decelerates to its master's velocity. In the velocity profiles below, the dotted lines indicate when the *Jog Plus* %Q bit is turned ON and then OFF.

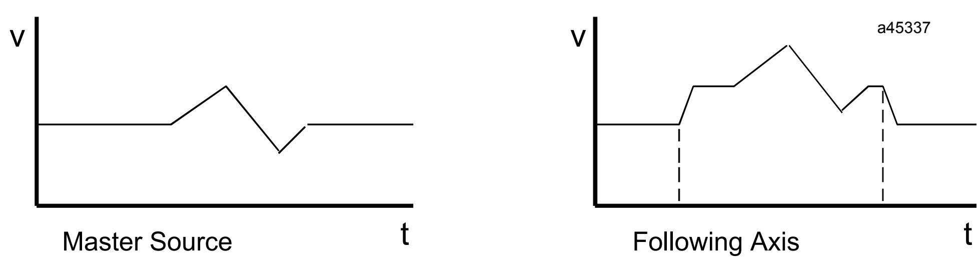


Figure 9-1. Combined Motion (Follower + Jog)

Follower Motion Combined with Motion Programs

Motion commands from stored programs or the *Move* %AQ command can also be combined with the master command to drive the follower axis. These point-to-point move commands can come from one of the stored motion programs 0 through 10 and any stored subroutines they may call. The *Move* %AQ command is treated as a single line motion program, which uses the present ***Jog Velocity*** and ***Jog Acceleration***. Program execution is started by the PLC setting an *Execute Program n* %Q bit or sending a *Move* %AQ command.

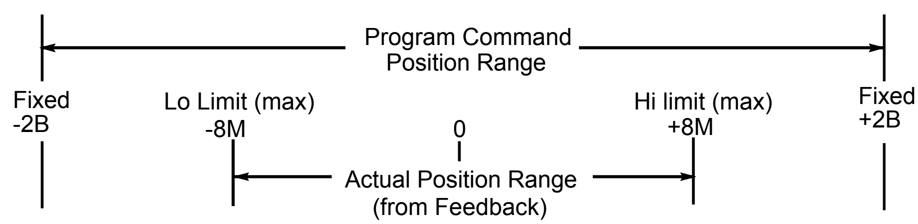
If there is no master command, the axis can be commanded solely from the stored motion program data. Thus, with no master input to Servo Axis 2 and Encoder 2 selected as the master source for Servo Axis 1, a stored program can be used to control Servo Axis 2 with Servo Axis 1 following per the designated ratio.

When PMOVEs are executed with Follower not enabled, the *In Zone* %I bit must be set at the end of the move before programmed motion will continue. When Follower is enabled, since *In Zone* may not turn on while also following a master command, the *In Zone* indication will not be required to continue. The next Move will take place when the commanded distance for the previous move has completed. The *In Zone* %I bit will always indicate the true in zone condition.

The active commanded position updated and used by the stored motion program is referred to as Program Command Position. Each time a program is selected for execution, this position register is initialized in one of the two ways listed below.

1. If the follower is **not enabled**, the Program Command Position is set to the current *Commanded Position = Actual Position + Position Error*.
2. If the follower **is enabled,** the Program Command Position is set to the Program Reference Position **(0).** Since the Program Command Position is only updated by internally generated commands (and not by the master command), it will then indicate the position commanded by the stored program. Absolute move commands from the stored program will be referenced to the Program Reference Position.

Position ranges (in counts) for the Actual and Program Command Position registers are indicated in the figure below.



With sustained commanded motion in the same direction, the Program Command Position will roll over at +2,147,483,647 or –2,147,483,648 counts

The Actual Position, however, will be confined by the configured high and low limits with maximum values of +8,388,607 and –8,388,608 counts.

Table 9-1 below indicates which source commands affect these position registers and the actual and commanded velocities. Program Command Position is updated only by internally generated move commands (program commands, *Jog Plus Minus*, *Find Home*, and *Move at Velocity*). The *Commanded Velocity* (returned in %AI data) also only indicates velocity commanded by these internally generated move commands. *Actual Position* and *Actual Velocity* %AI return data reflect the combination of the master input and the move commands. In other words, counts coming from the master source affect **only** the *Actual Position* and *Actual Velocity*. If there are no internally generated move commands, the *Commanded Velocity* will be 0 and the Program Command Position will not change.

Table 9-1. Command Input Effect on Position Registers

COMMAND Input	Follower Enabled	Follower Registers Affected by input
Master Commands (from selected Master source)	No	None affected
	Yes	*Actual Position* %AI status word is updated *Commanded Position* %AI status word is updated (*Actual Position* + *Position Error*) Program Command Position is **Not** affected *Actual Velocity* %AI status word is updated *Commanded Velocity* %AI status word is **Not** affected
Program Commands	No	*Actual Position* %AI status word is updated *Commanded Position* %AI status word is updated *Actual Position* + *Position Error*) Program Command Position is updated *Actual Velocity* %AI status word is updated *Commanded Velocity* %AI status word is updated (by Program commanded velocity only)
	Yes	*Actual Position* %AI status word is updated (by Program command + Master command) *Commanded Position* %AI status word is updated (*Actual Position* + *Position Error*) Program Command Position is updated (by Program command only) *Actual Velocity* %AI status word is updated (by Program command velocity + Master command velocity) *Commanded Velocity* %AI status word is Updated (by Program command velocity only)

Table 9-1. - Continued - Command Input Effect on Position Registers

COMMAND Input	Follower Enabled	Follower Registers Affected by input
Other Internally Generated Move Commands (Home, Jog, and Move at Velocity)	No	*Actual Position* %AI status word is updated *Commanded Position* %AI status word is updated (*Actual Position* + *Position Error*) Program Command Position is updated but not used *Actual Velocity* %AI status word is updated. *Commanded Velocity* %AI status word is updated (by Internal command velocity only)
	Yes (Find Home is not allowed)	*Actual Position* %AI status word is updated (by Internal command + Master command) *Commanded Position* %AI status word is updated (*Actual Position* + *Position Error*) Program Command Position is updated but not used *Actual Velocity* %A status word is updated (by Internal command velocity + Master command velocity) *Commanded Velocity* %AI status word is updated (by Internal command velocity only)

The Program Command Position can be synchronized to the *Actual Position* %AI value in 3 ways:

- *Find Home* %Q command execution
- *Set Position* %AQ command
- *Execute Motion Program n* %Q command (if the follower is not enabled)

The effect of these commands is indicated in Table 9-2 below.

Table 9-2. Actions Affecting Program Command Position

ACTION	Follower Enabled	Resulting Updates to Follower Position Registers
Home Found	No	*Actual Position* %AI status word is set to Home Value Program Command Position is set to *Actual Position* + *Position Error*
	Yes	*Find Home* %Q command is Not allowed Status Error is returned
Set Position %AQ Command	Not applicable	*Actual Position* %AI status word is set to %AQ Value Program Command Position is set to *Actual Position* + *Position Error* Note: Set Position is not allowed if the *Moving* %I bit is ON.
Execute Program	No	*Actual Position* %AI status word is NOT affected Program Command Position is set to *Actual Position* + *Position Error*
	Yes	*Actual Position* %AI status word is NOT affected Program Command Position is set to Reference Position (0)

Program moves will execute in a continuous fashion such that incremental PMOVE or CMOVE commands past the limits will roll over at the limit and continue. Absolute PMOVE or CMOVE commands can also be used for applications that do not require going beyond the high/low count limits.

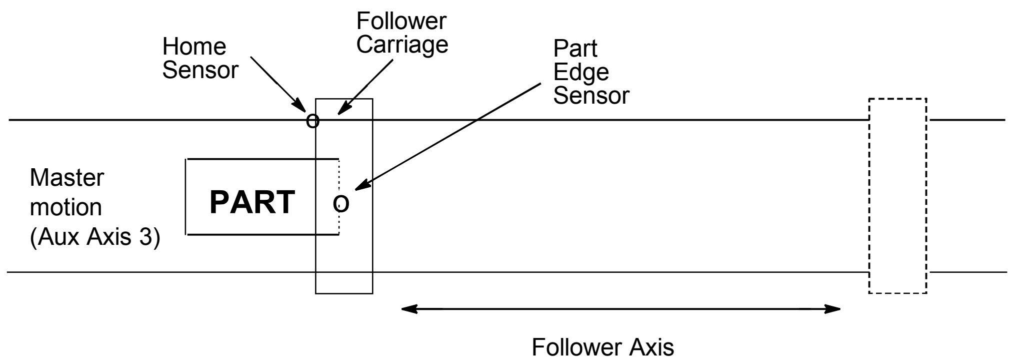
Any internally generated move command can be immediately terminated by the *Abort All Moves* %Q command.

The *User Selected Data* %AI status word can be changed to report the Program Command Position by using the *Select Return Data* %AQ command. Refer to Chapter 5 for details.

The following application example illustrates how a stored program can be used to control positioning operations relative to the detected edge of a moving object as it moves at a rate detected by the master axis (Aux Axis 3) encoder input.

Example 2: Follower Motion Combined with Motion Program

Applications that require modifying parts on the fly (such as notching, marking, riveting, spot welding, spot gluing, and so forth) would make use of the point-to-point moves superimposed on follower motion and enable follower at input features. A typical configuration and control sequence required for these applications is shown below.



Control Sequence:

1. With *Enable Follower* %Q bit OFF, the PLC commands Follower axis to home position where *Actual Position* & Program Command Position are synchronized and set to Home Position value. *Position Valid* %I bit indicates when this step is complete.

2. The PLC sets the *Enable Follower* %Q bit command.

Note

The CTL01- CTL16 bit to which the part edge sensor is connected would already have been configured in the ***Follower Enable Input*** configuration parameter.

3. When the Part edge sensor trips, the DSM302 enables the Follower axis to start following the master (Aux Axis 3) encoder inputs. The *Follower Enabled* %I bit indicates when the axis is

following the master command. Note that the Accel Ramp and Make-Up Time feature could be used to allow the follower axis to catch up to the master axis if required.

4. Once the follower is enabled, the PLC sends the *Execute Motion Program n* %Q bit to start execution of the selected program for the follower axis. At the time the program is selected, Program Command Position will be set to program reference position (0) because the follower is enabled. Program execution is then relative to the moving part edge as the follower axis tracks the part. Program Command Position now contains the position of the follower axis relative to the part edge and *Actual Position* indicates the total distance the follower axis has moved from the Home point (master +/– program commands).

5. At the end of program, the PLC turns *Enable Follower* %Q bit OFF and loops back to step 1 to repeat for next part.

Note

Since the DSM302 saved the Follower enable input trigger *Commanded Position* in a parameter register (#226 for axis 1, #234 for axis 2), step 1 this time could be used to execute another program with an absolute move command back to the parameter value position and continuing with step 2. In this case, the *In Zone* %I bit indication could be used to indicate when step 1 is complete.

This method is possible because the Program Command Position is set to the *Actual Position + Position Error* when execute motion program is commanded with the follower disabled.

Appendix A Error Reporting

DSM302 Error Codes

The DSM302 reports error codes in these %AI Table locations:

%AI Table Location	Data Reported	Usage
00	Module Status Code	Errors not related to a specific axis
01	Servo Axis 1 Error Code	Errors related to Servo Axis 1
02	Servo Axis 2 Error Code	Errors related to Servo Axis 2
03	Aux Axis 3 Error Code	Errors related to Aux Axis 3

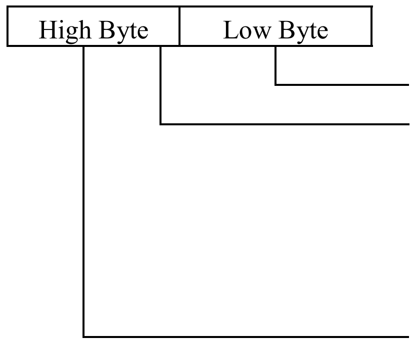
Each error code is a hexadecimal word which describes the error indicated when the *Module Error Present* %I status bit is set. The *Module Status Code* %AI status word reports module errors that are not related to a specific axis. Examples of such errors would be a self-test detected hardware failure or an error while storing a user program to the flash memory. All motion related errors are reported in the proper *Axis Error Code* %AI status word. Whenever the *Module Error Present* %I status bit is set, all four error words should be checked for a reported error.

Note

The STAT LED on the faceplate of the module flashes slow (four times/second) for Status Only errors and fast (eight times/second) for errors which cause the servo to stop. In the case of a fatal hardware error being detected at power-up, the STAT LED will flash an error code, which should be reported to GE Fanuc. See section "LED Indicators" later in this chapter for more details.

Error Code Format

All error codes are represented as hexadecimal data with the following format:



Bits 0-7	**Error Number (0-FFh)**
Bits 8-11 (low nibble)	**Axis Number** 0 - Axis Independent 1 - Servo Axis 1 2 - Servo Axis 2 3 - Aux Axis 3
Bits 12-15 (high nibble)	**Response Method** 0 - Status Only 1 - Stop Normal 2 - Stop Fast

Figure A-1. Status Code Organization

Response Methods

1. **Status Only Errors: Set** the *Module Error Present* %I bit and *Module Status Code* or *Axis Error Code* %AI word, but do not affect motion.

   Note

   Unless otherwise noted, any command which causes a Status Only Error is ignored.

2. **Stop Normal Errors: Perform** an internal abort of any current motion using current ***Jog Acceleration*** and ***Jog Acceleration Mode*** (LINEAR or S–CURVE). The *Drive Enabled* and *Axis Enabled* %I bit are each turned OFF after the configured ***Drive Disable Delay***.
3. **Stop Fast Errors: Instantly** abort all motion by setting the servo velocity command to zero. The *Drive Enabled* and *Axis Enabled %I* bits are each turned OFF after the configured ***Drive Disable Delay***.

Table A-1. DSM302 Error Codes

Error Number (Hexadecimal)	Response	Description	Error Type
`00`	None	No Error	All
		Configuration Errors	
`02`	Status Only	Scaled data too big, maximum value in range used	Axis
`03`	Status Only	Home Position > Positive EOT, Positive EOT used	Axis
`04`	Status Only	Home Position < Negative EOT, Negative EOT used	Axis
`05`	Status Only	Tuning parameter 1 invalid; data ignored	Axis
`06`	Status Only	Tuning parameter 2 invalid; data ignored	Axis
		Configuration Parameter Errors	
`10`	Status Only	Position Loop Time Constant too large, Immediate command ignored	Axis
`11`	Status Only	Position Loop Time Constant too small, Immediate command ignored	Axis
`12`	Status Only	Position Loop Time Constant computation overflow, reduced to non-overflow value	Axis
`1E`	Status Only	Immediate command Jog Velocity out of range, command ignored	Axis
`1F`	Status Only	Immediate command Jog Acceleration out of range, command ignored	Axis
		Program Errors	
`20`	Status Only	Program Acceleration overrange, defaults to 16.7 million cts/sec/sec	Axis
`21`	Status Only	Program Acceleration too small, defaulted to 32 cts/sec/sec	Axis
`22`	Status Only	Scaled Velocity greater than 1 million cts/sec, 1 million cts/sec is used	Axis
`23`	Status Only	Program Velocity is zero, defaulted to 1 count/sec used	Axis
`24`	Stop Normal	Program Position too large	Axis
`25`	Stop Normal	Unconditional Jump Destination not found	Axis
`26`	Stop Normal	Jump Mask error	Axis
`27`	Stop Normal	Wait Mask error	Axis
`28`	Stop Normal	Parameter Position too large	Axis
`29`	Status Only	Dwell time greater than 60 seconds, 5 seconds used	Axis
		Position Increment Errors	
`2C`	Status Only	Position Increment Overrange error, increment ignored	Axis
		Find Home Errors	
`30`	Status Only	Find Home while Drive Not Enabled error	Axis
`31`	Status Only	Find Home while Program Selected error	Axis
`32`	Status Only	Find Home while Force Digital Servo Velocity error	Axis
`33`	Status Only	Find Home while Jog error	Axis
`34`	Status Only	(1) Find Home while Move at Velocity error, or (2) Find Home while another Find Home Cycle is still active User executed the Find Home command (1) while executing a Move at Velocity (22h) AQ command or (2) while another Find Home cycle was in progress. For (1), halt the Move at Velocity operation (Moving I bit off) prior to executing the Find Home command. For (2), verify that axis is In Zone and not Moving before executing a Find Home command.	Axis
`35`	Status Only	Find Home While Follower Enabled	Axis
`36`	Status Only	Find Home while Abort bit set error	Axis
		Move at Velocity Errors	
`39`	Status Only	Move at Velocity while Drive Not Enabled error	Axis
`3A`	Status Only	Move at Velocity while Program Selected error	Axis
`3B`	Status Only	Move at Velocity while Home Cycle active error	Axis
`3C`	Status Only	Move at Velocity while Jog error	Axis
`3D`	Status Only	Move at Velocity while Abort All Moves bit is set error	Axis

Table A-1. - Continued - DSM302 Error Codes

Error Number (Hexadecimal)	Response	Description	Error Type
		Jog Errors	
40	Status Only	Jog while Find Home error	Axis
41	Status Only	Jog while Move at Velocity error	Axis
42	Status Only	Jog while Force Digital Servo Velocity error	Axis
43	Status Only	Jog while Program Selected and not Feedholding error	Axis
		Force Digital Servo Velocity Errors	
47	Status Only	Force Digital Servo Velocity while Jog error	Axis
48	Status Only	Force Digital Servo Velocity while Move at Velocity error	Axis
49	Status Only	Force Digital Servo Velocity while Program Selected error	Axis
4A	Status Only	Force Digital Servo Velocity while Follower Enabled error	Axis
		Set Position Errors	
50	Status Only	Set Position while Program Selected error	Axis
51	Status Only	Set Position Data overrange error	Axis
52	Status Only	Servo Axis 1,2: Set Position while not In Zone error Aux Axis 3: Set Position while ENC3 Velocity > 128 error	Axis
53	Status Only	Attempt to initialize position before digital encoder passes reference point.	Axis
54	Status Only	Digital encoder position invalid, must use Find Home or Set Position.	Axis
		End of Travel and Count Limit Errors	
56	Status Only	Commanded Position > Positive End of Travel or High Count Limit	Axis
57	Status Only	Commanded Position < Negative End of Travel or Low Count Limit	Axis
58	Status Only	(Absolute Position + Position offset) > Positive End of Travel or High Count Limit	Axis
59	Status Only	(Absolute Position + Position offset) < Negative End of travel or Low Count Limit	Axis
		Drive Disable Errors	
5B	Stop Normal	Drive Disabled while Moving	Axis
5C	Stop Normal	Drive Disabled while Program Active	Axis
		Software Errors	
5F	Status Only	Software Error (Call GE Fanuc Field Service)	Axis
		Program and Subroutine Errors	
60	Status Only	Absolute Encoder Rotary Position Computation error	Axis
61	Stop Normal	Subroutine not in list	Axis
62	Stop Normal	Call Error (subroutine already active)	Axis
63	Stop Normal	Subroutine End command found in Program	Axis
64	Stop Normal	Program End command found in Subroutine	Axis
65	Stop Normal	Sync subroutine encountered by non-sync program	Axis
		Program Execution Errors	
71	Status Only	Too many programs requested in same PLC sweep	Module
72	Status Only	Request Program 0-10 with multi-axis program active	Module
73	Status Only	Request two programs on same sweep with program active	Module
74	Status Only	Request two programs for same axis, lower number program executed	Module
75	Status Only	Empty or Invalid Program requested	Module
76	Status Only	AQ Move Command Position Out of Range	Axis

Table A-1. - Continued - DSM302 Error Codes

Error Number (Hexadecimal)	Response	Description	Error Type
3E	Status Only	Move at Velocity Data greater than 8,388,607 user units/sec	Axis
3F	Status Only	Move at Velocity Data greater than 1 million cts/sec error	Axis
Program Execution Conditions Errors			
80	Status Only	Execute Program while Home Cycle active	Axis
81	Status Only	Execute Program while Jog	Axis
82	Status Only	Execute Program while Move at Velocity	Axis
83	Status Only	Execute Program while Force Digital Servo Velocity	Axis
84	Status Only	Execute Program while Program Selected	Axis
85	Status Only	Execute Program while Abort All Moves bit set	Axis
86	Status Only	Execute Program while Position Valid not set	Axis
87	Status Only	Execute Program while Drive Enabled not set	Axis
88	Status Only	Execute Program with active Error Stop (Axis Enabled off)	Axis
Program Synchronous Block Errors			
8C	Status Only	Sync Block Error during CMOVE	Axis
8D	Status Only	Sync Block Error during Jump	Axis
EEPROM Errors			
90	Status Only	Flash EEPROM memory programming failure	Module
Hardware Limit Switch Errors			
A0	Stop Fast	Limit Switch (+) error	Axis
A1	Stop Fast	Limit Switch (–) error	Axis
Hardware Errors			
A8	Stop Fast	Out of Sync error	Axis
A9	Stop Fast	Encoder Loss of Quadrature or Linear Feedback Loss of Signal error	Axis
B0-BE	See Table A-2		
Encoder Alarms			
C0	Stop Fast	Servo not ready when MCON command is on - may be caused by the amplifier E-STOP input	Axis
C1	Status Only	Serial Encoder Battery Low	Axis
C2	Stop Normal	Serial Encoder Battery Failed	Axis
C3	Stop Normal	Servo Motor Over Temperature	Axis
C4	N/A	Not used.	N/A
C5	Stop Fast	Loss of Encoder	Axis
C6	Stop Fast	Error in encoder pulse detection	Axis
C7	Stop Fast	Encoder counter error	Axis
C8	Stop Fast	Encoder LED is disconnected	Axis
C9	Stop Fast	Encoder CRC checksum failure	Axis
CA	Stop Fast	Unsupported encoder, linear or Type A	Axis
CB	Stop Fast	Unsupported encoder, Type C	Axis
DSP Alarms			
D1	Stop Fast	Over current Detected	Axis
D2	Stop Fast	Loss of Analog Feedback	Axis
D3	Stop Fast	Over Acceleration Detected	Axis
D4	Stop Fast	Over Velocity Detected	Axis
D5	Status Only	KpVelFix Too Large	Axis
D6	Status Only	IntGainFix Too Large	Axis
D7	Status Only	Alpha Calculation Overflow G.S.	Axis
D8	Status Only	IntGainCur Calculation Overflow	Axis

D9	Status Only	KpCur Calculation Overflow	Axis
DA	Stop Fast	FPGA Error Detected	Axis

Table A-1. – Continued –DSM302 Error Codes

Error Number (Hexadecimal)	Response	Description	Error Type
Special Purpose Errors			
E0	Stop Fast	Custom Loop Type Mismatch	Axis
E2	Stop Fast	DSP Interrupt failure	Module
E9	Status Only	Set Internal Master Velocity command out of range – command ignored	Axis
Follower Ramp Errors			
EA	Status Only	Master velocity greater than 0.8*velocity limit-no distance compensation	Axis
EB	Stop Fast	Error in calculation during ramp-up	Axis
EC	Status Only	Programmed makeup time is not long enough for trapezoidal correction of the makeup distance	Axis
ED	Status Only	Velocity limit violation during ramp	Axis
EE	Status Only	Time limit violation during acceleration sector of the distance correction	Axis
Follower Errors			
F1	Status Only	Follower Position Error Limit Encountered	Axis
F2	Status Only	Follower Velocity Limit Condition Encountered	Axis
F3	Status Only	Follower Ratio B value = 0	Axis
F4	Status Only	Follower Ratio B value < 0	Axis
F5	Status Only	Follower Ratio A/B or B/A > 32	Axis
Winder Mode Errors			
F6	Status Only	A/B Change Not Allowed in Winder Mode With Follower Enabled	Axis
F7	Status Only	Set Winder Position Immediate Command Out of Zone	Axis
F8	Status Only	Zone Length Out of Range or Zone Length Change Exceeded 25%	Axis
F9	Status Only	Zone Length Change Not Accepted; Previous Change Still in Effect	Axis
FA	Status Only	Combination of follower B/A ratio and winder zone length exceeds limit (default winder zone length of 10000 used)	Axis
Internal Errors			
FD	Stop Fast	System software error	Axis
FE	Stop Fast	Unrecognized encoder, not supported	Axis

DSM Digital Servo Alarms (B0–BE)

GE Fanuc α and βdigital servo systems have built in detection and safety shut down circuitry for many potentially dangerous conditions. The table below reflects that three different models of servo amplifiers may be used with the DSM, the β Series, the α Series SVU and the α Series SVM. The following table indicates alarms supported by a particular servo amplifier and the corresponding DSM error code. *Table entries that are blank in the amplifier columns indicate amplifier alarms not supported by the particular amplifier series.* To clear a servo alarm, amplifier power cycle reset is required. Additionally, a "Clear Error " %Q discrete command is required to clear the DSM Error Code. Amplifier alarms not cleared by power cycle of the amplifier will continue to be reported to the DSM module. A brief trouble shooting section for servo alarms appears at the end of the error alarm tables.

Table A-2. DSM Digital Servo Alarms

Error Number (Hexadecimal)	Servo Alarm Name	Description	Amplifier Alarm Display		
			SVM 7 SEG	SVU 7 SEG	β ALM LED
B0	HV	Over- Voltage DC LINK	07†	1	ON
B1	LV	Low Voltage Control Power	06†	2	
B2	DBRLY	Dynamic Brake Circuit Failure † SVM PSM DC LINK Low Charge	 05†	7	
B3	LVDC	Low Voltage DC LINK	04†	3	ON
B4	OH	Amplifier Over Heat	03†		ON
B5	FAL	Cooling Fan Failure	02†		ON
B6		† SVM PSM IPM Alarm or Over Current	01†		
B7	DCSW DCOH	Regenerative Circuit – Failure Alarm Regenerative Circuit – Discharge Alarm	08†	4 5	ON
B9	LV5V	SVM Servo Module +5 V Low	2		
BA	IPML IPMM IPMN IPMLM IPMMN IPMNL IPMLMN	IPM Over Current, High Temp or Low Volt (L axis, M axis, N axis, L & M axes, M & N axes, N & L axes or L & M & N axes)	8. 9. A. b. C. d. E.	8. 9. A. b. C. d. E.	
BB	LVDC	SVM Servo Module Low DC LINK	5		
BD	FAL	SVM Servo Module Fan Failure	1		
BE	HCL HCM HCN HCLM HCMN HCNL HCLMN	Abnormally High Motor Current (L axis, M axis, N axis, L & M axes, M & N axes, N & L axes or L & M & N axes)	8 9 A b C d E	8 9 A b C d E	ON

† The two segment display on the SVM power supply module (PSM) indicates power supply alarms.

Troubleshooting Digital Servo Alarms:

The guidelines below are intended to assist in isolating problems associated with various servo alarms. If the items below do not fit the case or resolve the alarm, replace the servo amplifier, or call GE Fanuc Hotline for support. The appropriate amplifier and motor, Maintenance Manual or Description Manual, will include more detailed trouble shooting procedures.

HV (High-voltage) Alarm: This alarm occurs if the high voltage DC level (DC LINK) is abnormally high.

1. The AC voltage supplied to the amplifier may be higher than the rated input voltage. The β Series amplifier, three-phase supply voltage should be between 200 VAC to 240 VAC.

2. The external regeneration resistor may be wired incorrectly. Carefully check the connections of the regeneration resistor to the amplifier. Check that the resistance of the regeneration resistor is within 20% of the rated value. Replace the regeneration unit if the resistance is out of tolerance.
3. The regeneration resistor may not be capable of dissipating excess generated voltage. Review the calculations for selecting the regenerative discharge unit and replace with a resistor of higher wattage rating as needed. Reducing acceleration values and position loop gains (larger value *Position Loop Time Constant*) will additionally reduce regenerated voltage levels.

LVDC (Low Voltage DC Link: This alarm occurs if the high voltage DC level (DC LINK) voltage is abnormally low.

1. The AC voltage supplied to the amplifier may be missing or lower in value than the rated input voltage. The β Series amplifier, three-phase supply voltage should be between 200 VAC to 240 VAC. Verify that the proper level of AC voltage is supplied to the line input (L1, L2 and L3) connections of the amplifier.

DCOH or DCSW (Regeneration Alarm): The DCOH alarm occurs if the temperature of the regeneration resistors is too high. The DCSW alarm indicates problems in the switching portion of the regeneration circuitry.

1. If the external regeneration resistor is *not* used check that the temperature sensor input to the amplifier is shorted or jumped. The β Series amplifier jumper T604 should be installed on connector CX11-6.
2. The external regeneration resistor may be wired incorrectly. Carefully check the connections of the regeneration resistor to the amplifier. Check that the resistance of the regeneration resistor temperature sensor is near zero ohms at room temperature. Replace the regeneration resistor if the temperature sensor indicates an open condition.
3. The regeneration resistor may not be capable of dissipating excess generated voltage. Review the calculations for selecting the regenerative discharge unit and replace with a resistor of higher wattage rating as needed. Reducing acceleration values and position loop gains (larger value *Position Loop Time Constant*) will additionally reduce regenerated voltage levels.

OH (Over-heat Alarm): The temperature of the amplifier heat sink is too high or motor temperature is excessive.

1. Ambient temperature may be too high, consider a cooling fan for the servomotor. GE Fanuc supplies fan kits for most FANUC motors.
2. The motor may be operating in violation of duty cycle restrictions. Calculate the amount of cooling time needed based on the duty cycle curves published for the particular motor.
3. The motor may be over loaded. Check for excessive friction or binding in the machine.
4. For all the above problems, allow ten minutes cooling of the amplifier with minimum or no motor loading then cycle amplifier power to reset.

FAL (Fan Alarm): The cooling fan has failed.

1. Check the fan for obstructions or debris. With amplifier power removed attempt to manually rotate the fan.
2. For SVM type amplifier systems the power supply module (PSM) and the servo amplifier module each include a cooling fan. The alarm code will indicate which unit failed.
3. Some amplifiers have field replaceable fan units. If a replacement fan unit is not available, replace the amplifier.

HC, HCL, etc. (High Current Alarm): Motor current is excessive. For α Series amplifiers the suffix (L, M, N, etc.) indicates which axis is in alarm

1. Motor power wiring (U, V and W) may be shorted to ground or connected with improper phase connections. Check the wiring and connections. Check the servomotor for shorts to motor frame. Replace the motor if shorted.
2. Improper motor type code may be configured or excessive values for tuning parameters. Confirm that the proper motor is configured and lower gain values.
3. The amplifier maintenance manual will describe the procedure for monitoring motor current signals (IR and IS). If the waveforms are abnormal replace the amplifier. If excessive noise is observed check grounds and especially the cable shield grounds for the command cable (K1) to the amplifier.
4. The motor may be operating in violation of duty cycle restrictions. Calculate the amount of cooling time needed based on the duty cycle curves published for the particular motor.
5. The motor may be over loaded. Check for excessive friction or binding in the machine.
6. For all the above problems, allow ten minutes cooling of the amplifier with minimum or no motor loading then cycle amplifier power to reset.

LV (Low Voltage Control Power Alarm): The control voltage used to operate the low-voltage circuitry in the amplifier is too low.

1. α Series SVU type amplifiers will be shipped with default jumpers to use a single phase of the 220 VAC power to the amplifier. Optionally the user may remove the jumpers and connect 220 VAC control power separately. Check that a minimum 200VAC is available on terminals L1C and L2C for default installation or on connector CX3 (Y Key) for separate control power.
2. Check the amplifier fuse. If the fuse is open replace with a new fuse after checking control power voltage. If the second fuse blows open, replace the amplifier.

DBRLY (Dynamic Brake Relay Failure): This alarm indicates that the contacts of the braking relay are welded together. Replace amplifier immediately.

IPML, IPMM, etc. (IPM Alarm): The Intelligent Power Module (IPM) is the high current switching device in the amplifier. The IPM can detect over-current, over-heat or low-voltage conditions in the power switching circuitry. The suffix (L, M, N, etc.) indicates which axis is in alarm

1 Motor power wiring (U, V and W) may be shorted to ground or connected with improper phase connections. Check the wiring and connections. Check the servomotor for shorts to motor frame. Replace the motor if shorted.

2 Improper motor type code may be configured or excessive values for tuning parameters. Confirm that the proper motor is configured and lower gain values.

3 The amplifier maintenance manual will describe the procedure for monitoring motor current signals (IR and IS). If the waveforms are abnormal replace the amplifier. If excessive noise is observed check grounds and especially the cable shield grounds for the command cable (K1) to the amplifier.

4 The motor may be operating in violation of duty cycle restrictions. Calculate the amount of cooling time needed based on the duty cycle curves published for the particular motor.

5 The motor may be over loaded. Check for excessive friction or binding in the machine.

6 For all the above problems, allow ten minutes cooling of the amplifier with minimum or no motor loading then cycle amplifier power to reset.

LED Indicators

There are seven LEDs on the DSM302 module which provide status indications. These LEDs are described below.

STAT Normally ON. FLASHES to provide an indication of operational errors. Flashes *slow* (four times/second) for Status-Only errors. Flashes *fast* (eight times/second) for errors which cause the servo to stop.

ON: When the LED is steady ON, the DSM302 is functioning properly. Normally, this LED should always be ON.

OFF: When the LED is OFF, the DSM302 is not functioning. This is the result of a hardware or software malfunction that will not allow the module to power up.

Flashing: When the LED is FLASHING, an error condition is being signaled.

Constant Rate, CFG LED ON:

The LED flashes slow (four times / second) for Status Only errors and fast (eight times / second) for errors which cause the servo to stop. The operational error code will be placed in one of the first four %AI status words and the *Module Error Present* %I status bit will be ON.

Constant Rate, CFG LED Flashing:

If the STAT and CFG LEDs both flash **together** at a constant rate, the DSM302 module is in boot mode waiting for a new firmware download. If the STAT and CFG LEDs both flash **alternately** at a constant rate, the DSM302 firmware has detected a software watchdog timeout due to a hardware or software malfunction.

Irregular Rate, CFG LED OFF:

If this occurs immediately at power-up, then hardware or software malfunction has been detected. The module will blink the STAT LED to display two error numbers separated by a brief delay. The numbers are determined by counting the blinks in both sequences. Record the numbers and contact GE Fanuc for information on correcting the problem.

OK The OK LED indicates the current status of the DSM302 module.

ON: When the LED is steady ON, the DSM302 is functioning properly. Normally, this LED should always be ON.

OFF: When the LED is OFF, the DSM302 is not functioning. This is the result of a hardware or software malfunction that will not allow the module to power up.

CFG This LED is ON when a valid module configuration has been received from the PLC. Flashes *slow* (four times/second) during the Motion Program Store function. Flashes *fast* (eight times/second) during the Write User RAM to EEPROM operation.

EN1 When this LED is ON, the servo drive for Servo Axis 1 is enabled.

EN2 When this LED is ON, the servo drive for Servo Axis 2 is enabled.

EN3 When this LED is ON, the *Force Analog Output* command for Aux Axis 3 is active.

EN4 When this LED is ON, the *Force Analog Output* command for Aux Axis 4 is active.

Appendix B

DSM Parameter Download Using the COMM_REQ Instruction

This appendix describes how to load DSM Parameter Memory with a Communications Request (Abbreviated COMM_REQ in this chapter) ladder instruction. An advantage of the COMM_REQ is that it can load up to 16 parameters per instruction compared to 2 parameters per PLC sweep for the Load Parameter Immediate Command. Also, multiple COMM_REQ instructions may be used for this purpose in a ladder program, but only one Load Parameter Immediate Command may be used per scan. Therefore, the COMM_REQ is most useful for loading several or many parameters, and the Load Parameter Immediate Command is most useful for loading one or two.

In general, a COMM_REQ is used in a Series 90-30 PLC ladder program to communicate with an intelligent module. However, this chapter only discusses the COMM_REQ as it applies specifically to a DSM module. The chapter is divided into four sections:

- Section 1: The Communications Request
- Section 2: The COMM_REQ Ladder Instruction
- Section 3: The COMM_REQ Command Block
- Section 4: Example of a DSM COMM_REQ

Section 1: The Communications Request

The Communications Request uses the parameters of the *COMM_REQ Ladder Instruction* and an associated *Command Block* to define the characteristics of the request. An associated *Status Word* reports the progress and results of each request.

Structure of the Communications Request

The Communications Request is made up of three main parts:

- The COMM_REQ Ladder Instruction
- The Command Block, which is a block of PLC memory (usually %R memory) that contains instructions and data for the COMM_REQ.

- The Status Word, which is one word of memory that status/error codes are written to.

The figure below illustrates the relationship of these parts:

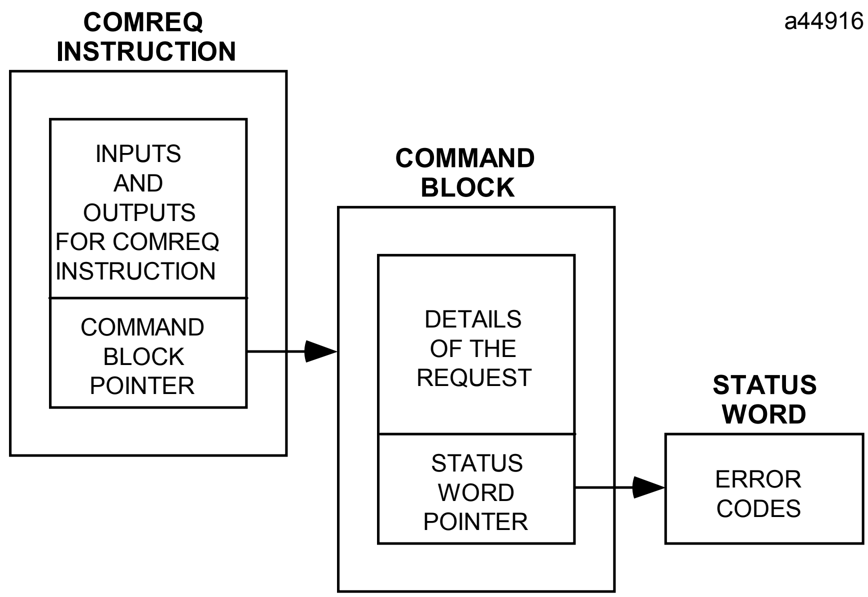


Figure B-1. Structure of the COMM_REQ

The COMM_REQ Ladder Instruction: The COMM_REQ Ladder Instruction is the main structure used to enter specific information about a communications request. This information includes the rack and slot location of the DSM module associated with the request, and a parameter that points to the starting address of the Command Block. Note that in programming this instruction, the command block data should be initialized in the ladder program before the rung containing the COMM_REQ instruction is executed.

The Command Block: The Command Block consists of several words of PLC memory that contain additional information about the communications request. This information includes timing parameters, a pointer to the Status Word, and a Data Block. The Data Block specifies the direction of the data transfer (via the Command Code) and location and type of data to be transferred.

The Status Word: The Status Word is a single location in PLC data memory where the CPU reports the result of the communications request. The Status Word address is specified in the Command Block by the user. The following table lists the status codes reported in the Status Word:

Table B-1. DSM COMM_REQ Status Word Codes

DSM COMM_REQ Status Word Codes			
Code Name	**Code #**	**Description**	**Possible Corrective Action**
IOB_SUCCESS	1	All communications proceeded normally.	None required.
IOB_PARITY_ERR	-1	A parity error occurred while communicating with an expansion rack.	Retry. Check hardware – expansion cables, DSM module, etc.
IOB_NOT_COMPL	-2	After the communication was over, the module did not indicate that it was complete.	Retry. Verify the COMM_REQ parameters.
IOB_MOD_ABORT	-3	The module aborted the communication.	Retry. Verify the COMM_REQ parameters.
IOB_MOD_SYNTAX	-4	The module indicated that the data sent was not in the correct sequence.	Verify the COMM_REQ parameters.
IOB_NOT_RDY	-5	The RDY bit in the module's status was not active.	Retry. Check DSM module.
IOB_TIMEOUT	-6	The maximum response time elapsed without receiving a response from the module.	Check DSM module. Verify the COMM_REQ parameters.
IOB_BAD_PARAM	-7	One of the parameters passed was invalid.	Verify the COMM_REQ parameters.
IOB_BAD_CSUM	-8	The checksum received from the DMA protocol module did not match the data received.	Retry. Check installation for proper grounding, shielding, noise suppression, etc.
IOB_OUT_LEN_CHGD	-9	The output length for the module was changed, so normal processing of the reply record should not be performed.	Verify the COMM_REQ parameters.

Corrective Action

The type of corrective action to take depends upon the application. If an error occurs during the startup or debugging stage of ladder development, the advice to "Verify the COMM_REQ parameters" is appropriate. The same is true if an error occurs right after a program is modified. But, if an error occurs in a proven application that has been running successfully, the problem is more likely to be hardware-related. The PLC fault tables should be checked for possible additional information when troubleshooting Status Word errors.

Monitoring the Status Word

Error Detection and Handling

As shown in the table above, a value of 1 is returned to the Status Word if communications proceed normally, but if any error condition is detected, a negative value is returned. If you require error detection in your ladder program, you can use a Less Than (LT) compare instruction to determine if the value in the Status Word is negative (less than zero). An example of this is shown in the following figure. If an error occurs, the Less Than's output (Q) will go high. A coil driven by the output can be used to enable fault handling or error reporting logic.

The FT output of the COMM_REQ, described later in this appendix, goes high for certain faults and can be used for fault detection also. Additionally, the Status Word can be monitored by error messaging logic for display on an Operator Interface device. In this case, certain Status Word codes would correspond to appropriate error messages that would display on the operator screen. For example, if a –1 were detected in the Status Word, a message could be displayed that says something like "Error communicating with the DSM module in expansion rack 2."

To dynamically check the Status Word, write a non-significant positive number (0 or 99 are typically used) into the Status Word each time before its associated COMM_REQ is executed. Then, if the instruction executes successfully, the CPU will write the number 1 there. This method lets you know that if the number 1 is present, it truly indicates that the last COMM_REQ executed successfully and that the 1 was not just "left over" from a previous execution. In the example presented at the end of this appendix, the number 99 is moved into the Status Word (%R0195) in a rung prior to the rung that contains the COMM_REQ instruction.

When multiple DSM COMM_REQs are used, it is recommended that each be verified for successful communications before the next is enabled. Monitoring the Status Word is one way to accomplish this.

Verifying that the DSM Received Correct Data

For critical applications, it may be advisable to verify that certain parameter values were communicated correctly to the DSM module before operation is allowed to continue. To accomplish this, first program the *Select Return Data* %AQ Immediate Command to specify a DSM parameter number to be read into the applicable *User Selected Data* %AI double word (there is one *User Selected Data* %AI double word for each axis). **Note that at least three PLC sweeps or 20 milliseconds, which ever represents more time, must elapse before the new *User Selected Data* is available in the PLC.** This requires programming some time delay logic to ensure that this requirement is met. Then, program a Double Integer Equal instruction to compare the value returned in the *User Selected Data* double word with the value sent. Section 4 of this appendix shows an example of this. Also, refer to Sections 2.10 and 4.21 of Chapter 5 for more information on the *User Selected Data* word and the *Select Return Data* command, respectively.

Operation of the Communications Request

The figure below illustrates the flow of information from the PLC CPU to the DSM module:

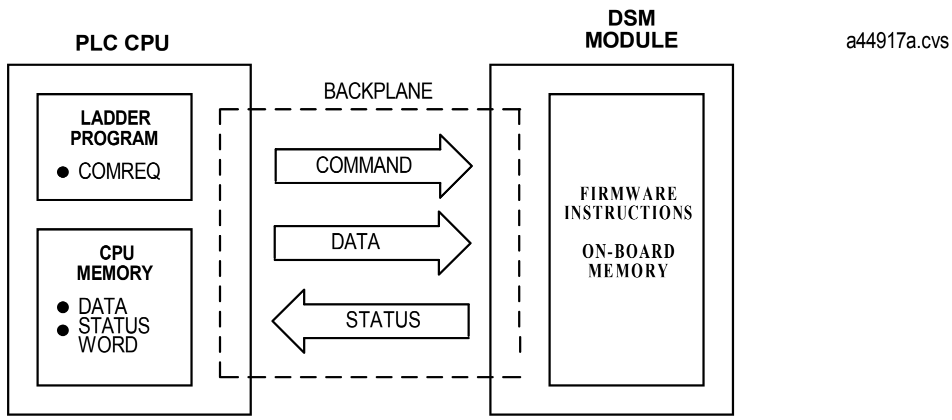


Figure B-2. Operation of the DSM Communications Request

A Communications Request is initiated when a COMM_REQ ladder instruction is activated during the PLC scan. At this time, details of the Communications Request, consisting of command and parameter data, are sent from the PLC CPU to the DSM module. The command data notifies the DSM that parameter data is to be sent, and directs the DSM to place the data into specified parameter memory locations. The order in which these instructions are sent is critical, so the Command Block should be programmed exactly as instructed later in this chapter. In the figure above, the DSM module is shown in the CPU rack and communications occur over the PLC backplane. If the DSM module is located in an expansion or remote rack, the commands and data are sent over the CPU rack's backplane, through the expansion or remote cable to the rack containing the DSM module, and across that rack's backplane to the DSM.

At the conclusion of every request, the PLC CPU reports the status of the request to the Status Word, which is a location in PLC memory that is designated by the Status Word Pointer in the Command Block.

Section 2: The COMM_REQ Ladder Instruction

The Communications Request begins when the COMM_REQ Ladder Instruction is activated. The COMM_REQ ladder instruction has four inputs and one output:

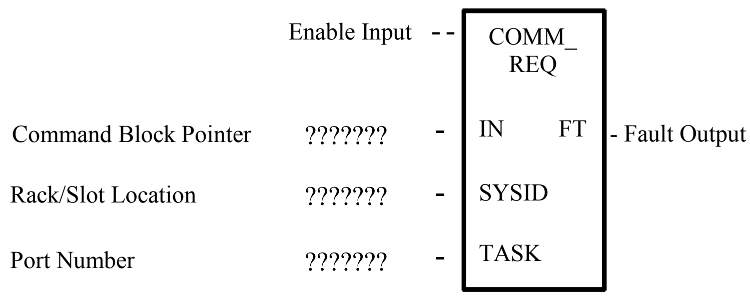


Figure B-3. DSM COMM_REQ Ladder Instruction

Each of the inputs and outputs is discussed in detail below. It is important to understand that the Command Block Pointer points to another location in memory where you must enter additional information about the Communications Request.

Enable Input: Must be Logic 1 to enable the COMM_REQ Instruction. It is recommended that the enabling logic be a contact from a transition ("one-shot") coil.

IN: The memory location of the first word of the Command Block. It can be any valid address within a word-oriented area of memory (%R, %AI, or %AQ).

SYSID: A hexadecimal value that gives the rack and slot location of the DSM module that the COMM_REQ is targeting. The high byte (first two digits of the hex number) contains the rack number, and the low byte contains the slot number. The table below shows some examples of this:

SYSID Examples

Rack	Slot	Hex Word Value
0	4	0004h
3	4	0304h
2	9	0209h

TASK: The number 0 should always be entered here for a DSM module.

FT Output: The function's FT (fault) output can provide an output to optional logic that can verify successful completion of the Communications Request. The FT output can have these states:

Table B-2. COMM_REQ Instruction FT Output Truth Table

FT Output		
Enable Input Status	**Does an Error Exist?**	**FT output**
Active	No	Low
Active	Yes	High
Not active	No execution	Low

- The FT output will be set High if:
  - The specified target address is not present (for example, specifying Rack 1 when the system only uses Rack 0).
  - The specified task number is not valid for the device (the TASK number should always be 0 for the DSM).
  - Data length is set to 0.

DSM COMM_REQ Programming Requirements and Recommendations

- It is recommended that DSM COMM_REQ instructions be enabled with a contact from a transition coil.
- If using more than one DSM COMM_REQ in a ladder program, verify that a previous COMM_REQ executed successfully before executing another one. This can be done by checking the Status Word and the FT (Fault) output, explained earlier in this appendix under the heading "Monitoring the Status Word."
- As seen in the table above, the FT output will be held False if the Enable Input is not active. This means that if the COMM_REQ is enabled by a transitional (one-shot) contact and a fault occurs, the FT output will only be High for one PLC scan. Therefore, to "capture" the fault, you can program the fault output as a Set coil, which would not be automatically reset at the end of a scan. Additional logic would then be needed to reset the fault output coil after the fault is acknowledged.
- Programming a device, such as a Set Coil, on the FT output of the COMM_REQ is optional.
- Note that the Series 90-30 COMM_REQ (unlike many of the other Series 90-30 PLC instructions) does not have an OK output.
- It is necessary to initialize the parameters in the Command Block prior to executing the COMM_REQ instruction. Since the normal PLC sweep order is from top to bottom, initializing the Command Block in an earlier rung (or rungs) than the rung that contains the COMM_REQ will facilitate this requirement. See the example at the end of this appendix.
- Recommendation: If you use MOVE instructions to load values into Command Block registers, use a Word-type MOVE to load a hexadecimal number, and an Integer-type MOVE to load a decimal number. You will see this applied in the example at the end of this appendix, where the E501h code is loaded via a Word-type MOVE instruction, and the remaining decimal values are loaded via Integer-type MOVEs.

Section 3: The COMM_REQ Command Block

The Command Block contains the details of a Communications Request. The first address of the Command Block is specified by the IN input of the COMM_REQ Ladder Instruction. This address can be in any word-oriented area of memory (%R, %AI, or %AQ). The Command Block structure can be placed in the designated memory area using an appropriate programming instruction (the BLOCK MOVE instruction is recommended).

The DSM Command Block has the following structure:

Table B-3. DSM COMM_REQ Command Block

COMM_REQ Command Block for DSM Module		
Description	**Address Offset**	**Word No. and Value**
Data Block Header Length	Address + 0	Word 1, always set to 4
WAIT/NOWAIT Flag	Address + 1	Word 2, always set to 0
Status Pointer Memory Type (see Memory Type Codes table, below)	Address + 2	Word 3, chosen by user (see Memory Type Codes table, below)
Status Pointer Offset	Address + 3	Word 4, chosen by user
Idle Timeout Value	Address + 4	Word 5, always set to 0
Maximum Communication Time	Address + 5	Word 6, always set to 0
Command Code	Address + 6	Word 7, always E501(hex.)
Parameter Data Size, in bytes	Address + 7	Word 8 (Size depends on the value in Word 12)
Parameter Data Memory Type	Address + 8	Word 9, chosen by user (see Memory Type Codes table, below)
Parameter Data Offset	Address + 9	Word 10, chosen by user
Starting parameter number (0 - 255)	Address + 10	Word 11, chosen by user
Number of parameters to load	Address + 11	Word 12, chosen by user
1st parameter data	Address + 12/13	Word 13 and Word 14
2nd parameter data	Address + 14/15	Word 15 and Word 16*
...	Address + …	...*
16th parameter data (4 bytes)	Address + 42/43	Word 43 and Word 44*

* Use of these words depends on the value in Word 12

Data Block Length (Word 1): The length of the Data Block header portion of the Command Block. It should be set to 4 for the DSM. The Data Block header is stored in Words 7 through 10 of the Command Block

WAIT/NOWAIT Flag (Word 2): This must always be set to logic **zero** for the DSM.

Status Word Pointer Memory Type (Word 3): This word specifies the memory type that will be used for the Status Word. Each memory type has its own specific code number, shown in the Memory Type Codes table below. So, for example, if you want to use %R memory for the Status Word, you would put either the decimal code number 8 or the hexadecimal code number 08h in this word.

Note that if you select a discrete memory type (%I or %Q), 16 consecutive bits will be assigned to the Status Word, beginning at the address specified in the Status Word Pointer Offset word, described below.

Status Word Pointer Offset (Word 4): This word contains the offset within the memory type selected. ***Note: The Status Word Pointer Offset is a zero-based number.*** In practical terms, this means that you should subtract one from the address number that you wish to specify. For example, to select %R0001, enter a zero (1 – 1 = 0). Or, if you want to specify %R0100, enter a 99 (100 – 1 = 99). Note that the memory type, %R in this example, is specified by the previous word (see the "Status Word Pointer Memory Type" explanation above).

Idle Timeout Value (Word 5): Since the DSM always uses the NOWAIT mode (WAIT/NOWAIT flag always set to zero), this Idle Timeout Value parameter is not used for the DSM. Set it to zero.

Maximum Communication Time (Word 6): Since the DSM always uses the NOWAIT mode (WAIT/NOWAIT flag always set to zero), this Maximum Communication Time parameter is not used for the DSM. Set it to zero.

Command Code (Word 7): This is always E501(hexadecimal) for the DSM. To enter this value directly as a hexadecimal value, use a Word-type MOVE instruction. Also, since this value is 58,625 in decimal, an Integer-type MOVE instruction (limited to a maximum decimal value of 32,767 because bit 16 is used for the sign) does not have the capacity to contain it. A Word-type MOVE instruction can hold a decimal number up to 65,535 (FFFF in hex.).

Parameter Data Size (Word 8): Specifies the Parameter Data size in bytes. This value depends on the value in Word 12, which specifies the number of parameters to be loaded. This value may be between 8 and 68. It is equal to 4 bytes (for the first two words of the Parameter Data section) plus 4 additional bytes for each parameter loaded. For example, if you wish to load 16 parameters (the maximum per COMM_REQ), multiply 4 times 16 to arrive at 64. Add 4 to 64 for a total of 68 bytes.

Parameter Data Memory Type (Word 9): This word specifies the memory type that will be used for Parameter Data. Each memory type has a unique code number, shown in the Memory Type Codes table below. So, for example, to specify %R memory, you would put either the decimal code number 8 or the hexadecimal code number 08h in this word.

Note that if you select a discrete memory type (%I or %Q), a group of 32 consecutive bits will be required for each parameter, and a group of 16 consecutive bits each will be required for Words 11 and 12.

Parameter Data Start Pointer Offset (Word 10): This word contains the offset within the memory type selected in the Parameter Data Memory Type parameter. ***Note: The Parameter Data Pointer Offset is a zero-based number.*** In practical terms, this means that you should subtract one from the address number that you wish to specify. For example, to select %R0001 as the Parameter Data Start location, enter zero (1 – 1 = 0). Or, to select %R0100, enter 99 (100 – 1 = 99). Note that the memory type, %R in this example, is specified in the previous word.

Starting Parameter Number (Word 11): Specifies the number of the first parameter to be loaded. Valid values are 0 – 255. However, this number must take into account the value in Word 12. For example, if Word 12 specifies that 10 parameters are to be loaded, the Starting Parameter Number must be less than 247; otherwise, the number of the last parameter to be loaded would be out of range (would be greater than 255).

Number of Parameters to Send (Word 12): Specifies how many parameters will be loaded. Valid values are 1 – 16. The value in this word impacts the value of Word 8.

Parameter Data (Words 13 - 44): The size of this Parameter Data area depends on the value in Word 12 (Number of Parameters to Send). Two words (4 bytes) of data are required for each parameter. Since the valid number of Double Integer parameters is 1 through 16, the Parameter Data area can be between 2 and 32 words.

COMM_REQ Memory Type Codes: The codes in the following table are used in Word 3 (Status Word Pointer Memory Type), and Word 9 (Parameter Data Memory Type).

Table B-4. COMM_REQ Memory Type Codes

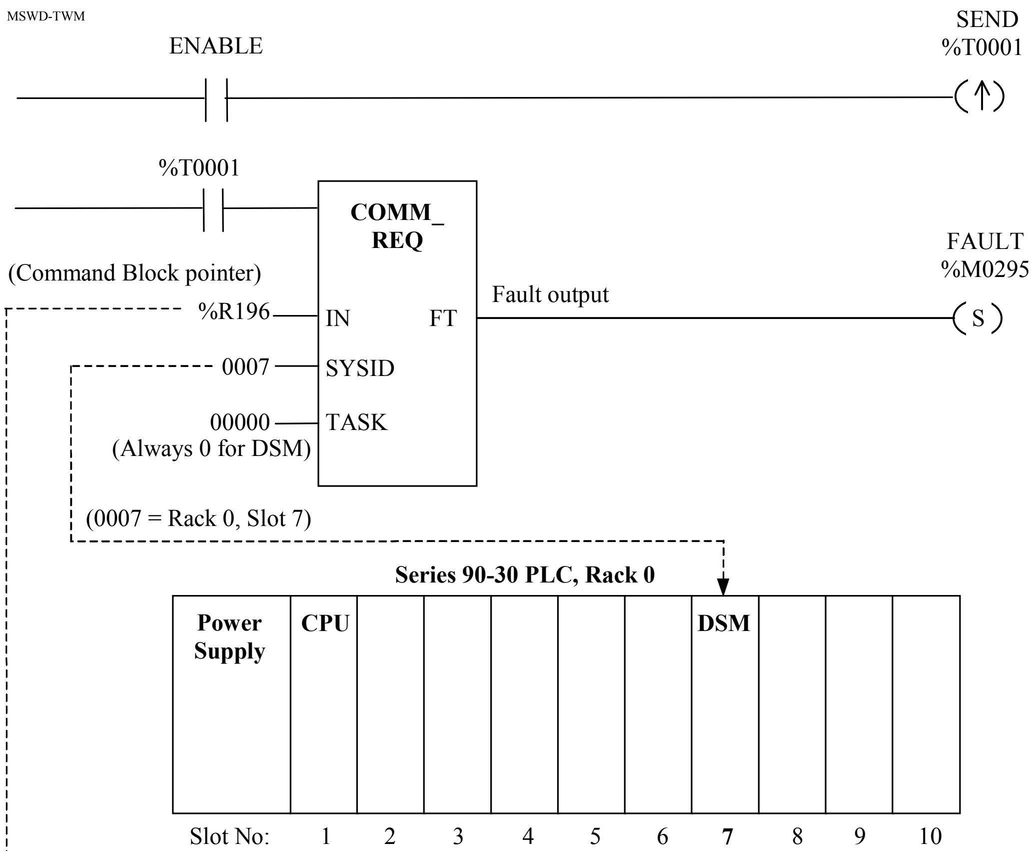
COMM_REQ Memory Type Codes			
Memory Type Abbreviation	Memory Type	Code Number to Enter	
		Decimal	Hexadecimal
%I	Discrete input table	70	46h
%Q	Discrete output table	72	48h
%R	Register memory	8	08h
%AI	Analog input table	10	0Ah
%AQ	Analog output table	12	0Ch

Section 4: DSM COMM_REQ Example

In this example, the following specifications are given:

- The DSM module is mounted in Rack 0, Slot 7 of the PLC.
- The Command Block's starting address is %R0196.
- The Status Word is located at %R0195.
- 16 parameters are to be sent.
- The COMM_REQ's FT (fault) output drives a Set Coil.
- DSM Parameter 1 is considered critical in this application. The last two rungs of the example verify that Parameter 1 received the correct value via the COMM_REQ.

The next figure provides an overview of the COMM_REQ example:



Command Block for DSM COMM_REQ		
Memory Address	**Value**	**Description**
%R196	4	Length, in words, of Data Block header (always 4)
%R197	0	WAIT/NOWAIT Flag. Always 0 for DSM
%R198	8	8 = Register memory (%R) for Status Word address
%R199	194	Specifies register 195 for Status Word address
%R200	0	Always 0 for DSM
%R201	0	Always 0 for DSM
(Start of "Data Block" section of Command Block)		
%R202	E501	Always E501 for DSM
%R203	68	Parameter Data size, in bytes
%R204	8	8 = Register memory (%R) for Data Block Parameter Data
%R205	205	Specifies register 206 for start of Parameter Data
(Start of "Parameter Data" section of Data Block)		
%R206	1	Starting Parameter Number
%R207	16	Number of Parameters to send
%R208 - %R239	xxxxx	Parameter data for 16 contiguous double word parameters.

%R195: Status Word Register

Data Block Header (4 words)

Data Block Parameter Data (68 bytes)

Figure B-4. Overview of the DSM COMM_REQ Example

```
|   (**********************************************************************)
|   (*  THE BLKMV (BLOCK MOVE) LOADS THE COMM_REQ HEADER DATA.  NOTE        )
|   (*  THAT %R0196 - %R0201 ARE IMPLIED ADDRESSES IN THE BLKMV INSTRUCTION )
|   (**********************************************************************)
|                                                                    SEND
| LOAD                                                              %T0001
+--] [-------------------------------------------------------------(↑)
|  SEND
| %T001                           +-----+
+--] [---------------------------+BLKMV+-
|                                 | INT |
|                                 |     | STATUS
|                         CONST -+IN1 Q+-%R0195
|                         +00099 |     |
|                                |     |
|                         CONST -+IN2  |
|                         +00004 |     |
|                                |     |
|                         CONST -+IN3  |
|                         +00000 |     |
|                                |     |
|                         CONST -+IN4  |
|                         +00008 |     |
|                                |     |
|                         CONST -+IN5  |
|                         +00194 |     |
|                                |     |
|                         CONST -+IN6  |
|                         +00000 |     |
|                                |     |
|                         CONST -+IN7  |
|                         +00000 +-----+
|
|
|   (**********************************************************************)
|   (* PUT THE COMMAND TYPE, E501 (hex), IN THE FIRST DATA WORD, R202.     *)
|   (* PUT THE BYTE COUNT OF THE DATA IN THE NEXT WORD (R203).  PUT THE    *)
|   (* MEMORY TYPE OF THE DATA (8=%R) IN THE NEXT WORD (R204). PUT THE     *)
|   (* STARTING LOCATION OF THE DATA BLOCK IN THE NEXT WORD (R205).        *)
|   (* PUT THE STARTING PARAMETER NUMBER (1) IN THE LOWER BYTE OF (R206)   *)
|   (* AND THE NUMBER OF PARAMETERS TO BE SENT (16) IN THE LOWER BYTE      *)
|   (* OF R207.  NOTE:  THE FIRST MOVE INSTRUCTION SHOULD BE A WORD TYPE   *)
|   (* TO FACILITATE ENTERING THE HEX VALUE (E501). THE REST OF THE MOVE   *)
|   (* INSTRUCTIONS SHOULD BE INTEGER TYPES TO FACILITATE ENTERING DECIMAL *)
|   (* VALUES.                                                             *)
|   (**********************************************************************)
|
```

```
|  SEND
| %T0001  +-----+                +-----+                 +-----+
+--] [----+MOVE_+----------------+MOVE_+-----------------+MOVE_+-
|         | WORD|                | INT |                 | INT |
|         |     |                |     |                 |     |
|         |     | CMDTYP         |     | BYTECNT         |     | MEMTYP
|  CONST -+IN  Q+-%R0202  CONST -+IN  Q+-%R0203   CONST -+IN  Q+-%R0204
|   E501  | LEN |         +00068 | LEN |          +00008 | LEN |
|         |00001|                |00001|                 |00001|
|         |     |                |     |                 |     |
|         +-----+                +-----+                 +-----+
|
|  SEND
| %T0001  +-----+                +-----+                 +-----+
+--] [----+MOVE_+----------------+MOVE_+-----------------+MOVE_+-
|         | INT |                | INT |                 | INT |
|         |     |                |     |                 |     |
|         |     | DATAST         |     | PAR_NO          |     | NO_VALS
|  CONST -+IN  Q+-%R0205  CONST -+IN  Q+-%R0206   CONST -+IN  Q+-%R0207
|  +00205 | LEN |         +00001 | LEN |          +00016 | LEN |
|         |00001|                |00001|                 |00001|
|         |     |                |     |                 |     |
|         +-----+                +-----+                 +-----+
|
|
|  (************************************************************************)
|  (* ADD LOGIC HERE TO MOVE THE PROPER DATA INTO THE REGISTERS            *)
|  (* (%R208 - %R239) SO THEY CAN BE SENT TO THE DSM302 PARAMETERS. FOR    *)
|  (* EXAMPLE, THE VALUE IN %R208/%R209 WILL BE SENT TO PARAMETER 1.       *)
|  (* (NOTE THAT THE PARAMETERS ARE DOUBLE INTEGER SIZE.)  LIKEWISE, THE   *)
|  (* VALUE IN %R210/%R211 WILL BE SENT TO PARAMETER 2 AND SO ON, UNTIL    *)
|  (* FINALLY, THE VALUE IN %R238/%R239 WILL BE SENT TO PARAMETER 16.      *)
|  (************************************************************************)
|
```

In order to facilitate loading single precision registers (16 bits) with double-integer parameter values (32 bits), you could use the Multiply DINT instruction to initialize the PLC registers that will be sent to the DSM302. There are several possible advantages to this approach:

- This is an easy way to convert single word registers to double-integer form, thus simplifying your number handling in the ladder program.
- It gives flexibility if you need to scale the values. Since User Unit scaling usually implies a decimal point (i.e. 1000=1.000 inch), you could, for example, use the DINT Multiply instruction and a constant of 1000. This would let you specify an inch as 1 instead of 1000 on your Operator Input device.
- Many low cost Operator Interface devices cannot write a double integer value. Multiplying the single register value from the Operator Interface in a double integer instruction produces a double integer result.

In the example below, the single word value from an Operator Input device will be scaled by a factor of 1000, then placed into the double integer word %R208/R209 (for Parameter 1).

```
|%I0001  +-----+                 +-----+
+--] [---|BCD4_+-----------------+ BLK +-
|        | TO_ |                 | CLR |
|        | INT |                 |LEN 1|
|%Q0017 -|IN  Q+-%R0150   %R0151-|IN   |
| +00021 +-----+  +00021         +-----+
|
|%I0001          +-----+
+--] [-----------| MUL_+-
|                | DINT|
|                |     |
|        %R0150 -|I1  Q+-%R0208
|    +0000000021 |     | +0000021000
|                |     |
|         CONST -|I2   |
|    +0000001000 +-----+
```

In the example above, the BCD value in %I0017 (from a BCD Operator Input device), is converted to an integer value and placed in %R0150 when switch %I0001 is closed. Also, %R0151 is cleared to zero. Note that on the output of the BCD4-TO_INT instruction, %R0150 is shown as a single word integer value (displayed as a signed, 5-digit number). However, when %R0150 is used in the second rung as the input to the double integer Multiply instruction (MUL_DINT), the CPU automatically combines it with the next %R address (%R0151) to form a double-integer word (displayed as a signed, ten-digit number).

When used this way, %R0151 is called an “implied address” since it is not shown on the screen. Be aware that you must not use %R0151 for any other purpose, (it should be held to a value of zero); otherwise, the value placed into %R0150 in the first rung would be corrupted. The same principle applies in the case of double word %R0208/R0209. Here, the use of %R209 is implied by the fact that %R0208 is displayed as a double integer word. So %R0209 should be reserved for this use only.

In the second rung, the value in %R0150 is multiplied by 1000. This provides scaling of the input value by a factor of 1000. If scaling were not desired, the value 1 would be used instead.

```
|  (**********************************************************************)
|  (* NOW ACTIVATE THE COMM_REQ TO SEND THE PARAMETER DATA TO THE DSM302  *)
|  (**********************************************************************)
|  SEND                                                               FAULT
| %T0001   +-----+                                                    %M0295
+--] [-----+COMM_|    |------------------------------------------------(S)
|          | REQ |    |
|          |     |    |
|  HDR_WDS |     |    |
|  %R0196 -+IN FT+---|
|          |     |
|   CONST -+SYSID|
|    0007  |     |    NOTE:  SYSID HIGH BYTE = COMM_REQ RACK DESTINATION
|          |     |           SYSID  LOW BYTE = COMM_REQ SLOT DESTINATION
|   CONST -+TASK |           TASK ALWAYS = 0 FOR DSM302 COMM_REQ
| 00000000 +-----+
```

Verifying the Data Sent to Parameter 1

In this example, the value in DSM Parameter 1 is critical because it specifies a move distance that, if incorrect, could result in machine damage. So, the logic in the following two rungs verifies that the Parameter 1 value is correct. If not correct, contacts (not shown) from output coil “VERIFY” in the second rung will prevent the DSM from producing motion.

```
|%M0200  +-----+                 +-----+                 +-----+      %M0202
+---][---|MOVE_+-----------------|MOVE_+-----------------| TMR +--------( )--
|        | WORD|                 | INT |                 |.001s|
|        |     |                 |     |                 |     |
| CONST -|IN  Q+-%AQ001   CONST -|IN  Q+-%AQ002   CONST -|PV   |
|   1840 | LEN | 1840     +00001 | LEN | +00001   +00045 |     |
|        |00001|                 |00001|                 +-----+
|        +-----+                 +-----+                 %R0300
|                                                        +00000
|%M0202         +-----+                                               VERIFY
+---][----------| EQ_ |+----------------------------------------------( )--
|               | DINT||
|               |     ||
|        %R0208 -|I1  Q++
|    +0000015000 |     |
|               |     |
|        %AI0021-|I2   |
|    +0000015000 +-----+
```

First Rung: The MOVE_WORD instruction moves hexadecimal number 1840 into %AQ0001, the first word of the Immediate Command. The low byte value (40h) specifies the *Select Return Data* Immediate Command. The high byte value (18h) specifies the Mode selection for *Parameter Data.*

The MOVE_INT instruction moves a decimal value of 1, indicating Parameter 1, into %AQ0002. This commands that the value in DSM Parameter 1 be written to the *User Selected Data* double word for Axis 1, %AI0021/AI0022 in this example. Note: The actual %AI addresses used for any DSM module are specified when the module is configured.

The TMR timer instruction produces a 45-millisecond time delay after the *Select Return Data* Immediate Command is sent. This is required because *User Selected Data* is not available in the ladder until at least 3 sweeps or 20 milliseconds (which ever is greater) elapses after the *Select Return Data* Immediate Command is sent. Since the sweep time in this example is 14 milliseconds, this 45-millisecond delay ensures that the Parameter 1 data will be present in the *User Selected Data* double word before the Equal instruction in the next rung executes. Note that contact %M0200 must stay ON long enough for the TMR timer to time out and enable the second rung.

Second Rung: After the 45-millisecond delay in the previous rung elapses, contact %M0202 closes and enables this rung. In this rung, a double integer EQUAL instruction compares the value in %R0208/R0209 (the source of the value sent by the COMM_REQ to DSM Parameter 1) with the value returned in %AI0021/AI0022. If the values are equal, the output coil "VERIFY" will turn on.

Appendix C Position Feedback Devices

There are four GE Fanuc α and β Series Digital serial encoder models that will function with the DSM302:

8K	(8,192 cts/rev)	- No longer available on new motors
32K	(32,768 cts / rev)	- Standard on β Series motors
64K	(65,536 cts/rev)	- Standard on α Series motors
1000K	(1,048,576 cts/rev)	- Optional on α Series motors

Table C-1. Digital Serial Encoder Resolutions

Note

The older "A" or "C" Series million count serial encoder will not operate with the DSM302. An error will be reported if this encoder is connected.

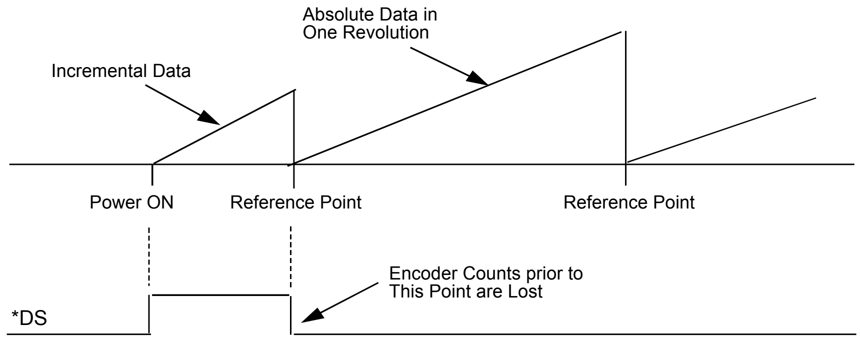
For position control purposes, by default, the DSM302 treats all encoders as 8192 counts/rev. The additional resolution of 32K, 64K and 1000K encoders will still be used in the digital servo velocity controller to provide smooth operation at low speeds. To use the increased position feed back resolution, available in release 1.3 or later firmware, refer to the Tuning Parameter section of the Configuration Chapter.

Digital Serial Encoder - First Time Use or Use After Loss of Encoder Battery Power

The encoder temporarily provides incremental data during the first use or after restoring encoder battery power. The incremental data is lost when motor shaft rotation causes the encoder to pass a reference point (similar to a marker signal) within one revolution of the motor shaft. The diagram below describes this situation, as well as the state of the *DS internal signal. The signal *DS is not directly available to the user but serves to illustrate the sequence of events. The *DS signal indicates that absolute position data is available. If no battery pack is used, *DS will be reset on each power cycle.

Note: The asterisk in *DS indicates that it is a negated or inverse form of signal DS. Therefore, if DS is logic 1, *DS is logic 0; and if DS is logic 0, *DS is logic 1. Sometimes an overline is used instead of an asterisk to indicate this relationship:

$$*DS = \overline{DS}$$



Note

The GE Fanuc Digital serial encoder must be rotated up to one full revolution after the absolute mode battery has been reattached to the amplifier. Within one revolution the encoder will reference itself and report a referenced status (*DS = 0) to the DSM302. The encoder will set its internal counts accumulator to zero at the point where *DS switches to a logic zero. Counts accumulated prior to the *DS transition are lost.

Digital Serial Encoder Modes

The GE Fanuc Digital serial encoders can be operated in either Incremental mode or Absolute mode. The mode is configured using the ***Feedback Mode*** selection in the configuration software. Proper operation of the Absolute mode requires an external battery pack that must be connected to the servo amplifier. Refer to the appropriate amplifier manual for selection and installation of the battery pack.

Limitations on Total Travel for Linear Axis Mode

When ***Axis Mode*** is set to LINEAR in the configuration software, the POS EOT and NEG EOT configuration values set the limits of programmed motion. The maximum value that can be used for the *+EOT* is +8,388,607 user units. User units are configured as a ratio to the encoder count value and are limited to the range of 8:1 to 1:32. Additionally, the position feedback from a digital servo is, by default, fixed at 8192 counts per revolution. A simple formula can be used to determine the maximum number of motor shaft revolutions possible for axis travel when Linear (non-continuous) ***Axis Mode*** is used. If the position feedback resolution is changed via Tuning Parameter 1, then substitute the new resolution.

$$\text{Maximum Absolute Revolutions} = \frac{8{,}388{,}608}{\text{Position Counts Per Rev}} * \frac{\text{User Units}}{\text{Counts}}$$

Using the above calculation and the 8192 position counts per revolution default, the maximum possible move is realized with the ***User Units*** to ***Counts*** ratio at the maximum 1:32 resulting in a

maximum travel of +/– 32,768 motor shaft revolutions. (The maximum travel originates at the *Actual Position* zero and can move the number of maximum travel revolutions plus or minus of the zero position.)

There is no restriction on maximum travel for CONTINUOUS mode. Refer to Chapter 4 for additional information on LINEAR and CONTINUOUS Mode.

Incremental Encoder Mode Considerations

The digital serial encoder can be used as an incremental encoder returning 8192 counts per shaft revolution, with no revolution counts retained through a power cycle. The equivalent of a *marker* pulse will occur once each motor shaft revolution. All ***Home Modes*** (HOMESWitch, MOVE+, Move–) and *Set Position* %AQ commands reference the axis, and set the *Position Valid* %I bit upon successful completion. The configured ***High Count Limit*** and ***Low Count Limit*** are valid and the A*ctual Position* %AI status word as reported by the DSM302 will wrap from high to low count or from low to high count values. This is an excellent mode for continuous applications that will always operate via incremental moves, in the same direction. ***Home Offset*** and ***Home Position*** configuration items allow simple referencing to the desired location.

Absolute Encoder Mode Considerations

The GE Fanuc Digital serial encoder can be used as an absolute type encoder by adding a battery pack to retain servo position while system power is off. A Find Home cycle or *Set Position* %AQ command must be performed initially or whenever encoder battery power is lost with the servo amplifier also in a powered down state. ***Feedback Mode*** set to ABSOLUTE must be selected in the configuration software for proper operation with a battery pack.

Absolute Encoder Mode - Position Initialization

When a system is first powered up in Absolute Encoder mode, a position offset for the encoder must be established. Using the *%Q Find Home* cycle or the *Set Position % AQ* command can accomplish this.

Find Home Cycle - Absolute Encoder Mode

The ***Find Home Mode*** can be configured for MOVE PLUS (+), MOVE MINUS(–) or HOMESWitch operation. Refer to Chapter 4 for additional details of Home Cycle operation. The ***Home Offset*** and ***Home Position*** configuration items function the same as in Incremental Encoder mode. At the completion of the Home Cycle, the *Actual Position* %AI status word is set to the configured ***Home Position*** value. The DSM302 internally calculates the encoder Absolute Feedback Offset needed to produce the configured ***Home Position*** at the completion of the Home Cycle. This Absolute Feedback Offset is immediately saved in the DSM302 non-volatile (capacitor backup) memory.

Once an absolute position is established by successful completion of a Find Home cycle, the DSM302 will automatically initialize the *Actual Position* %AI status word after a power cycle and set the *Position Valid* %I bit.

Set Position Command - Absolute Encoder Mode

The *Set Position* %AQ command functions the same way as in incremental encoder mode. At the completion of the *Set Position* operation, *Actual Position* is set to the *Set Position* value. The DSM302 internally calculates the encoder Absolute Feedback Offset needed to produce the commanded *Set Position* value. This Absolute Feedback Offset is immediately saved in the DSM302 non-volatile (capacitor backup) memory.

Once an absolute position is established by a *Set Position* command, the DSM302 will automatically initialize *Actual Position* after a power cycle and set the *Position Valid* %I bit.

Absolute Encoder Mode - DSM302 Power-Up

The battery pack attached to the servo subsystem will maintain power to the encoder counter logic. Once the encoder has referenced through first time start up, the actual position is automatically maintained by the encoder, even if the axis is moved during servo power loss. The encoder will monitor the status of the battery pack, and report loss of battery power or low battery power to the DSM302.

The DSM302 will complete a power-on diagnostic, and when configured for absolute encoder mode, interrogate the referenced status of the Digital serial encoder. A valid referenced status from the encoder will signal the DSM302 to read the encoder absolute position. The DSM302 will report the *Actual Position* %AI status as the sum of the encoder position and the Absolute Feedback Offset established by the initial Find Home cycle or *Set Position* %AQ command.

Absolute Encoder Mode with Continuous Mode

Some restrictions are necessary when Absolute Encoder mode is selected along with ***Axis Mode*** set to CONTINUOUS.

Absolute Encoder mode causes the DSM302 to automatically initialize the *Actual Position* %AI status word from the battery backed absolute encoder after a power cycle. Absolute Encoder mode is selected by setting ***Feedback Mode*** to ABS in the configuration software.

CONTINUOUS axis mode allows the DSM302 to create continuous motion in one direction by the use of multiple CMOVE or PMOVE incremental commands. As the axis moves, *Actual Position* will reach a ***High*** or ***Low Count Limit*** then roll over to the other limit. The distance between *Actual Position* rollovers is the Continuous Count Modulus.

The battery backed absolute encoder has a maximum total absolute counting range of (8192 cts/rev) x (32767 revs) = 268,427,264 counts. See the calculation in the previous section when using higher resolution position feedback. This is the maximum number of counts the encoder can move after a position reference operation (*Find Home* %Q or *Set Position* %AQ command) and still retain absolute position after a power cycle. If the encoder rotates more than this number of counts after a position reference operation, the absolute position "rolls over" and the encoder starts a new counting cycle.

Restrictions when Absolute Encoder Mode is used with Continuous Mode

1. If the Continuous Count Modulus (in counts) is a power of 2, then no restrictions exist on continuous travel. This means the distance defined as (***Hi Count Limit*** – ***Low Count Limit***) + 1 in counts must be a number which is a power of 2 such as 128, 256, 512 ... 8192, 16384, and so forth. Under this condition the DSM302 will always initialize the *Actual Position* %AI status word to the correct continuous position after a power cycle.
2. If the power of 2 condition for Continuous Count Modulus is not met, the continuous motion must be limited to a range of +/– 268,427,264 counts (using the default 8192 CountsPerRev for the position feedback) after a *Set Position* %AQ or *Find Home* %Q position reference operation. This restriction can be handled in some systems by periodically performing a *Set Position* when the axis is stopped and holding a known continuous position.

Note

Performing a *Set Position* %AQ command or *Find Home* %Q cycle in ABSOLUTE encoder mode causes the DSM302 to recalculate the encoder Absolute Feedback Offset and automatically save the offset data in non-volatile memory.

Incremental Quadrature Encoder

Incremental Quadrature Encoders provide three output signals to the DSM302: Channel A, Channel B, and Marker. The Channel A and Channel B signals transition as the encoder turns, allowing the DSM302 to count the number of signal transitions and calculate the latest encoder position change and direction of rotation.

Incremental Quadrature Encoders are incremental feedback devices; they do not provide a continuous indication of absolute shaft angle as the input shaft rotates. For this reason, the DSM302's *Actual Position* %AI status word must be initialized with a known physical position before positioning control is allowed. This position alignment can be accomplished using the *Set Position* %AQ Immediate command or the *%Q Find Home* cycle . The home cycle makes use of the encoder marker channel, which is a once per revolution pulse produced at a known encoder shaft angle. Successful completion of the %Q *Find Home* cycle or a *Set Position* %AQ command causes the DSM302 to set the axis *Position Valid* %I bit. *Position Valid* must be set before motion programs will be allowed to execute. *Position Valid* is only cleared by an encoder Quadrature Error (Channel A and Channel B switching at the same time) or by turning on the *Find Home* and *Abort* %Q bits simultaneously.

Note

In Digital Mode, only incremental quadrature encoders are supported for the Follower mode master axis.

Appendix D Start-Up and Tuning GE Fanuc Digital and Analog Servo Systems

This appendix provides a procedure for starting up and tuning a GE Fanuc Digital or Analog servo system. For Digital servos systems, there are two control loops in the DSM302 that require tuning, the velocity loop and the position loop. Always begin with module configuration then proceed to the velocity loop setting and finally the position loop. For Analog servo systems, there are a series of Start-Up Procedures to follow.

GE Fanuc Start-Up and Tuning Information for DIGITAL Servo Systems

There are three major sections covered:

- Validating Home Switch, Over Travel Inputs and Motor direction.
- Tuning the Velocity Loop.
- Tuning the Position Loop.

Validating Home Switch, Over Travel Inputs and Motor direction

1. Connect the motor, amplifier and DSM302 module following the procedures in Chapter 2.
2. If Over travel Limit switches are used (OT Limit SW = ENABLED in configuration), wire them to the correct 24V terminal board points (refer to Chapter 3). The overtravel inputs are operated in the fail-safe mode i.e. a normally closed or PNP type switching device should be used. Current must be sourced to the input to maintain a logic level 1 on the input while the axis is NOT at the EOT position or an alarm condition (Error A9) will be returned. Otherwise the Over travel limits, ***Pos EOT*** and ***Neg EOT***, must be disabled using the Logicmaster configuration software.
3. If a Home switch is used (Home Mode = HOMESW in configuration), wire it to the correct 24V terminal board points (refer to Chapter 3). The Home switch must be wired and actuated so that it is ALWAYS ON (closed) when the axis is on the negative side of home and ALWAYS OFF (open) when the axis is on the positive side of home. Typically the Home switch is mounted at or near one end of the axis travel. It is important to verify the operation of

the home switch prior to attempting a home cycle. It may be necessary to reverse the motor direction (Motor1 or Motor2 Dir = POS/NEG) in the module configuration.

4. Use the configuration software to set the desired user scaling factors and other configurable parameters. The following items MUST be changed from the default configuration settings:

Configuration Item	Setting
Feedback Type:	Digital
Servo Cmd:	Digital
Motor Type:	Select from Table in Chapter 4
Position Loop Time Constant:	60 ms
Velocity Loop Gain:	(Load Inertia / Motor Inertia) * 16
User Units : Counts (Standard Mode Only)	See Chapter 3
Position Error Limit:	30000 x User Units / Counts
Velocity at 10v:	139820 x User Units / Counts

Set the configuration parameters in the order shown above. Changing ***Feedback Type*** or ***Control Loop*** type will cause **all** other DSM configuration changes to be erased and set back to the default values. If a Follower Control Loop is used be sure to set the ***AI/AQ Length*** to 50/9 so that the additional Aux Axis 3 %AI data will be reported to the PLC.

5. Store the configuration to the PLC.

6. Clear the program from the PLC, turn off all DSM302 %Q bits and place the PLC in RUN mode. Monitor the %I CTL bits for Home Switch, (+) Overtravel and (-) Overtravel and confirm that each bit responds to the correct switch (Refer to Chapter 5 for %I bit definitions).

7. Turn on the *Enable Drive* %Q bit and confirm that the servo amplifier is enabled. If a brake is used on the servomotor it should be released at this time.

8. Send the %AQ command code for *Force Digital Servo Velocity* 100. Confirm that the motor moves in the desired POSITIVE direction and the *Actual Velocity* reported in the %AI table is POSITIVE. If the motor moves in the wrong direction, use the Motor Dir parameter in the configuration software to swap the positive and negative axis directions.

9. Remove the *Force Digital Servo Velocity* command from the %AQ table. Use a low speed Jog velocity and Jog acceleration in the configuration, values may be increased later. Turn on the *Jog Plus* %Q bit. Confirm that the servo moves in the proper direction and that the *Actual Velocity* reported by the DSM302 in the %AI table matches the configured ***Jog Velocity.***

10. Use a low speed value for Find Home velocity and Final Home velocity in the module configuration, values may be increased later. Check for proper operation of the Find Home cycle by momentarily turning on the *Find Home* %Q bit (the *Drive Enabled* %Q bit must also be maintained on). The axis should move towards the Home Switch at the configured ***Find Home Velocity***, then seek the Encoder Reference point at the configured ***Final Home Velocity***. If necessary, adjust the configured velocities and the location of the Home Switch for consistent operation. The final Home Switch MUST transition at least 10 milliseconds before

the encoder reference point is encountered. The physical location of ***Home Position*** can be adjusted by changing the ***Home Offset*** value with the configuration software.

11. Monitor servo performance and use the *Jog Plus* and *Jog Minus* %Q bits to move the servomotor in each direction. Placing the correct command code in the % AQ table can temporarily modify the *Position Loop Time Constant*. For most systems the ***Position Loop Time Constant*** can be reduced until some servo instability is noted, then increased to a value approximately 50% higher. Once the correct time constant is determined, the DSM302 configuration should be updated using the configuration software. ***Velocity Feedforward*** can also be set to a non-zero value (typically 90 – 100 %) for optimum servo response. Refer to *Tuning a GE Fanuc Digital Servo* for information on setting the digital servo ***Velocity Loop Gain***.

12. If Follower mode is used with an Incremental Quadrature Encoder, confirm that *Actual Position* (Aux Axis 3) represents the encoder position. **Make sure the desired Follower axis slave : master ratio has been programmed as the A:B ratio using the configuration software**.

Digital Servo System Startup Troubleshooting Hints

1. The DSM302 requires PLC firmware release 6.50 or greater and Logicmaster 90-30/20/Micro software release 8.02 or greater.

2. The default DSM302 configuration for the ***Overtravel Limit Switch*** inputs is ENABLED. Therefore, 24 vdc must be applied to the Overtravel inputs or the DSM302 will not operate. If Overtravel inputs are not used, the DSM302 configuration should be set to ***Overtravel Limit Switch*** inputs DISABLED.

If the *Axis Enabled* %I bit is OFF, the axis will not respond to any %Q bits or %AQ commands. When a servomotor is not used with a Servo Axis, the ***Motor Type*** must be set to 0 or *Axis Enabled* will stay OFF. A ***Motor Type*** of 0 disables the axis servo loop processing and sets *Axis Enabled ON*, allowing the axis to accept commands such as *Load Parameter Immediate* and *Set Analog Output Mode*.

3. **The *Enable Drive* %Q control bit must be set continuously to ON or no motion other than Jogs will be allowed.** If no STOP errors have occurred, the *Drive Enabled* %I status bit will mirror the state of the *Enable Drive* %Q bit. A STOP error will turn off *Drive Enabled* even though *Enable Drive* is still ON. The error condition must be corrected and the *Clear Error* %Q control bit turned ON for one PLC sweep to re-enable the drive.

4. If the *Module Error Present* %I status bit is ON and the *Axis Enabled* and *Drive Enabled* %I status bits are OFF, then a STOP error has occurred (Status LED flashing fast). **In this state, the Servo Axis will not respond to any %Q bits or %AQ commands other than the *Clear Error* %Q bit.**

5. The *Clear Error* %Q control bit uses one-shot action. Each time an error is generated, the bit must be set OFF then ON for at least one PLC sweep to clear the error.

6. The CFG OK LED must be ON or the DSM302 will not respond to PLC commands. If the LED is OFF then a valid DSM302 configuration has not been received from the PLC.

Tuning a GE Fanuc Digital Servo Drive

The following pages provide you with an introduction to the basics required for tuning a GE Fanuc Digital servo drive. This introduction shows one method for tuning a servo drive. The method will not work in all applications, and you should modify the approach based on the application. In order to display and measure the necessary signal waveforms, the DSM302 analog outputs must be connected to an oscilloscope. Without an oscilloscope to measure the signals, tuning the servo drive with the following approach will not be possible. The *Select Analog Output Mode* %AQ command (47h) is used to select the data that is sent to the analog outputs during servo tuning. Refer to Chapter 5 for a discussion of this %AQ command.

Tuning Requirements

The module has three main parameters that are adjusted during tuning. The parameters are the ***Position Loop Time Constant****,* ***Velocity Feed Forward Gain****,* and ***Velocity Loop Gain****.* The ***Position Loop Integrator Time Constant*** gives the position loop an additional degree of freedom but in typical applications is not required.

The approach to tuning the control loops is to tune the inner control loops first. In this example, the inner control loop that requires tuning is the velocity loop. As shown in the figure below, the position loop is the outer loop and sends velocity commands to the velocity loop.

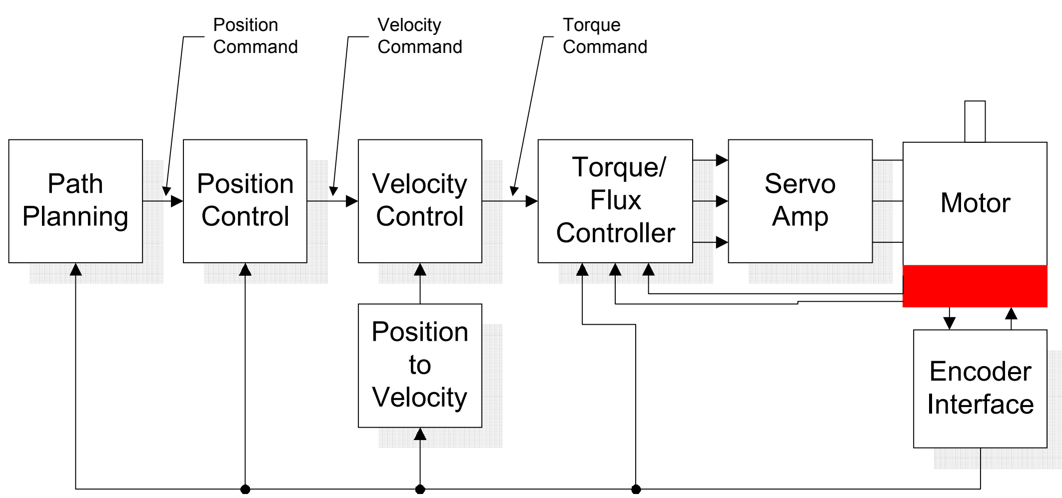


Figure D-1. Control Loops Block Diagram

Tuning the Velocity Loop

The proper method to tune the velocity loop is to separate the velocity loop from the position loop. To achieve this separation, a method must be used to directly send velocity commands without using the position loop control. The DSM module has several modes that will allow the user to send a velocity command directly to the velocity loop. Two methods are as follows:

Method #1:

The *Force Digital Servo Velocity* %AQ immediate command (34h) will send a velocity command directly to the velocity loop. This command is different from the *Move at Velocity Command,* which uses the position loop to generate the command. This is important since we do not want the position loop interacting with the velocity loop at this point in our tuning process. The *Force Digital Servo Velocity* %AQ command allows the user to generate a step change in the velocity. The velocity command step is then used to generate the velocity loop step response. The user should note that when a velocity command step change is performed the acceleration is limited only by the bandwidth of the velocity loop. In some applications this can cause damage to the controlled device due to the high acceleration rate.

Method #2:

In some applications, method #1 introduces too large a shock to the device under control. In these cases, another method to generate a velocity command is needed. The method requires that the user set the position loop to an open loop configuration. The position loop is set to open loop by setting the ***Position Loop Time Constant*** to zero and the ***Velocity Feedforward Gain*** to 100 percent. You can then use the *Move at Velocity Command* or a motion program to generate velocity commands to the servo drive. The first parameter that needs to be adjusted is the ***Velocity Loop Gain***. The parameter adjusts the velocity loop bandwidth. As a starting point use the following formula (also reference the Velocity Loop Gain Section):

Equation 1

$$\text{Velocity Loop Gain} = \frac{J_l}{J_m} \cdot 16$$

Where:

J_l = Load Inertia

J_m = Motor Inertia

The ***Velocity Loop Gain*** calculated in equation 1 in many cases will not need to be altered. However, due to the application (for example, machine resonance) the value may need to be adjusted. To tune the ***Velocity Loop Gain*** the following procedure can be used:

1. Choose the method to introduce velocity command to the velocity loop. Method #1 and Method #2 (above) are examples of methods to perform this task.

2. Connect an oscilloscope to the analog outputs for Motor Velocity feedback and Torque Command. See Section 4.25 of Chapter 5 for analog output configuration instructions.
3. Set the Velocity Loop Gain to zero. This is a conservative approach. If the application is known to not have resonant frequencies from zero to approximately 250 Hz, you can start with a higher value, but do not exceed the value calculated in equation 1 at this point.
4. Generate a velocity command step change. At this point the step change should be relatively small compared to the full speed of the machine. Ten to 20 % of the rated machine speed is a good start.
5. Observe the Motor Velocity and Torque Command on the oscilloscope. The objective is to obtain a critically damped velocity loop response. Pay particular attention to any oscillations that are occurring in the velocity feedback signal.
6. Increase the ***Velocity Loop Gain*** in small steps and repeat 4 and 5 until instability in the Motor Velocity feedback signal is observed. Once this point is reached, decrease the ***Velocity Loop Gain*** by at least 15 %. As a general rule, the lower the ***Velocity Loop Gain*** value that meets the system requirements the more robust the control. You should carefully observe the velocity feedback signal. In some applications, running the ***Velocity Loop Gain*** high enough to create instability can cause machine damage. If in doubt, adjust the ***Velocity Loop Gain*** to be no greater than the value calculated in equation 1. If oscillations are observed in the Motor Velocity feedback signal prior to this point, decrease the ***Velocity Loop Gain*** and continue with step 7 below.
7. The velocity loop is tuned at this point. However, the robustness of the loop must be checked. To perform this test, introduce velocity command steps in increments of 20% Rated Machine Speed, 40% Rated Machine Speed, 60% Rated Machine Speed, 80% Machine Rated Speed, and 100% Rated Machine Speed. Observe the Motor Velocity and Torque Command signals for any instability. If an instability or resonance is observed, reduce the ***Velocity Loop Gain*** and repeat the test.

NOTE: For Digital servos, the %AQ Force Analog Output command can provide Torque Command or Commanded Motor Velocity. (Velocity = 750 rpm/volt and %TqCmd = (100/1.111111 Volt)*X Volt or Torque Cmd = 100% Torque Command = 1.111 Volts, 100%TqCmd = MaxCur Amplifier. For instance: Beta 0.5 MaxCurAmp = 12 amps => 1.111111Volt = 12 amps.

Sample Velocity Loop Tuning Session

A sample velocity loop tuning session is shown in the plots that follow.

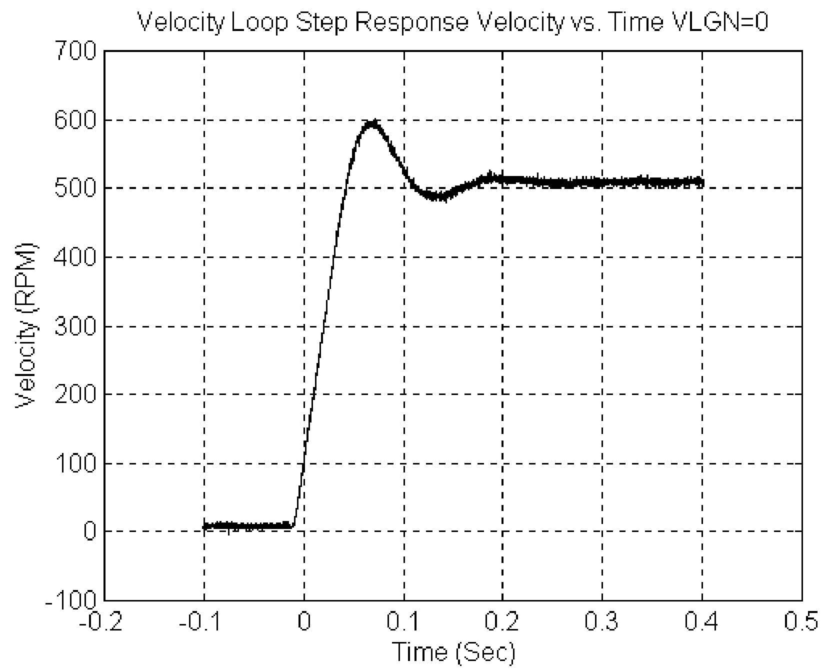


Figure D-2. Velocity Loop Step Response Velocity vs. Time VLGN = 0

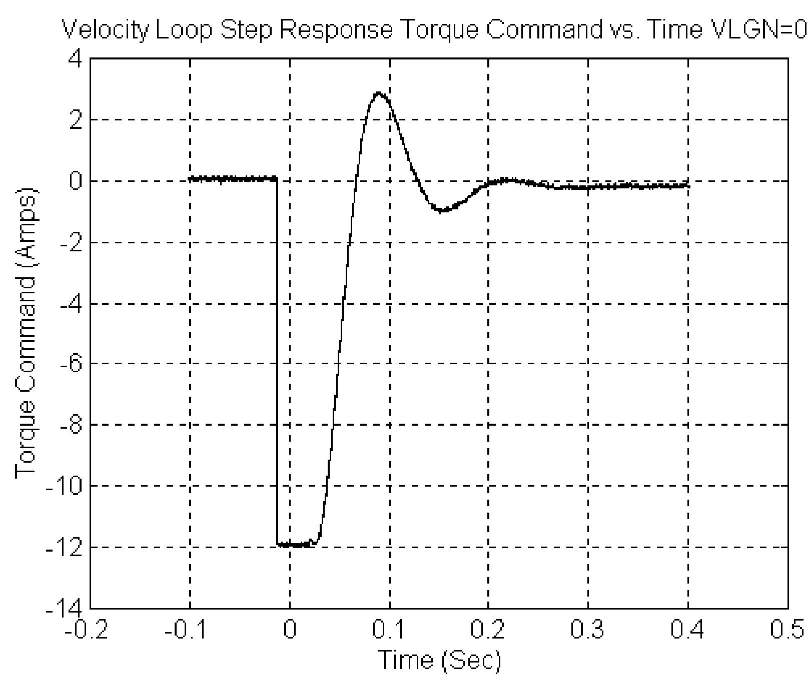


Figure D-3. Velocity Loop Step Response Torque Command vs. Time VLGN = 0

Note that in Figures D-2 and D-3 the system does not have enough damping. In this case the controller does not have the required bandwidth and the ***Velocity Loop Gain*** must be increased.

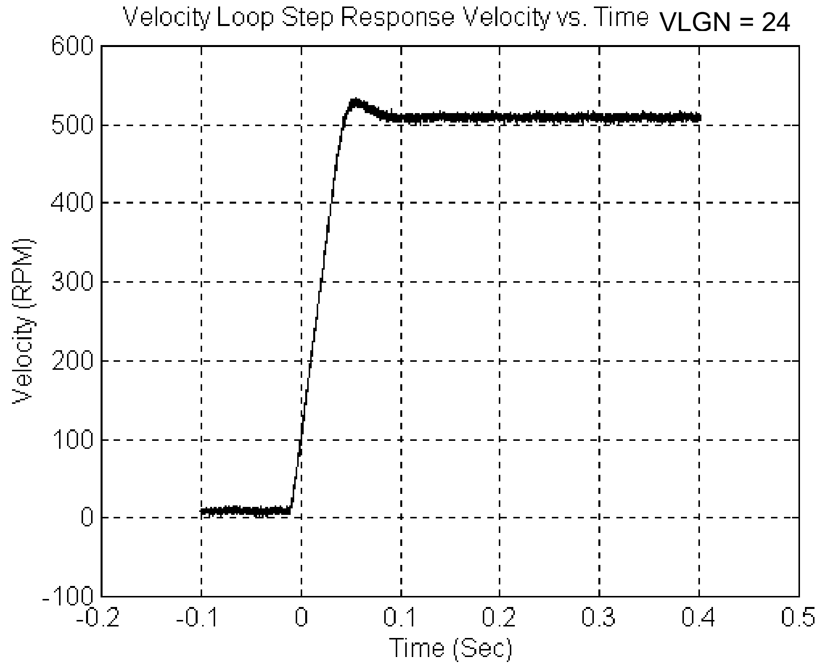


Figure D-4. Velocity Loop Step Response Velocity vs. Time VLGN = 24

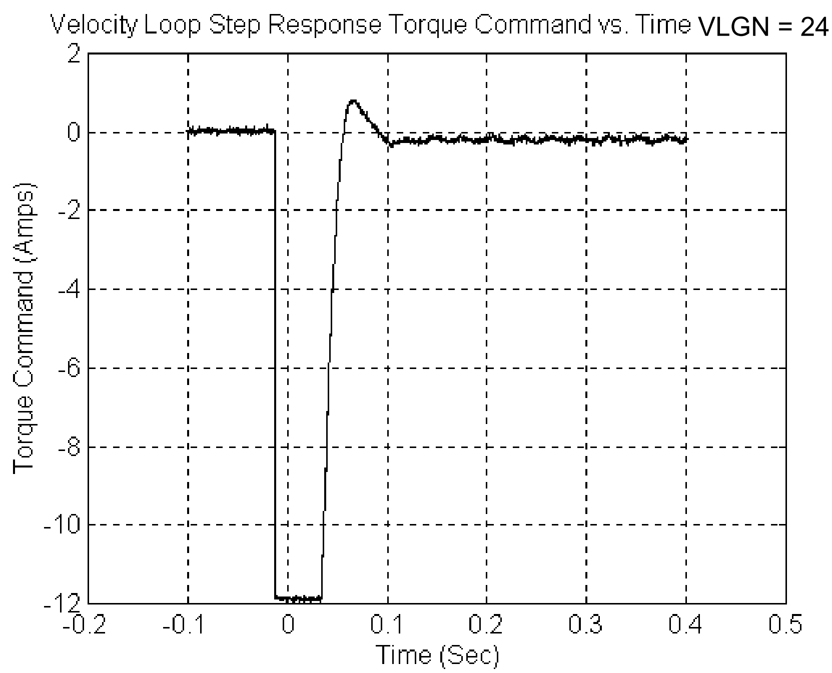


Figure D-5. Velocity Loop Step Response Torque Command vs. Time VLGN = 24

Note that in Figures D-4 and D-5, the system is beginning to look acceptable. The only problem is the velocity overshoot.

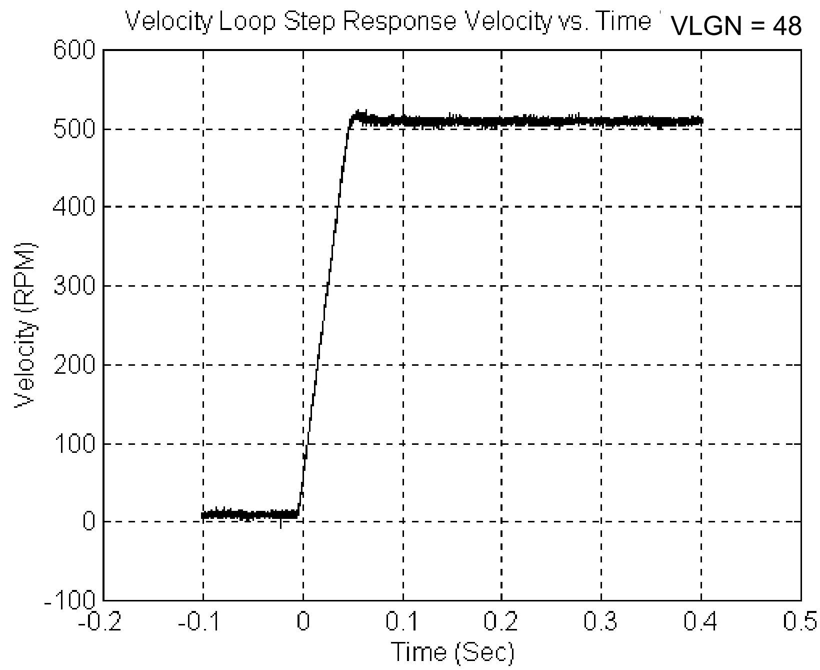


Figure D-6. Velocity Loop Step Response Velocity vs. Time VLGN = 48

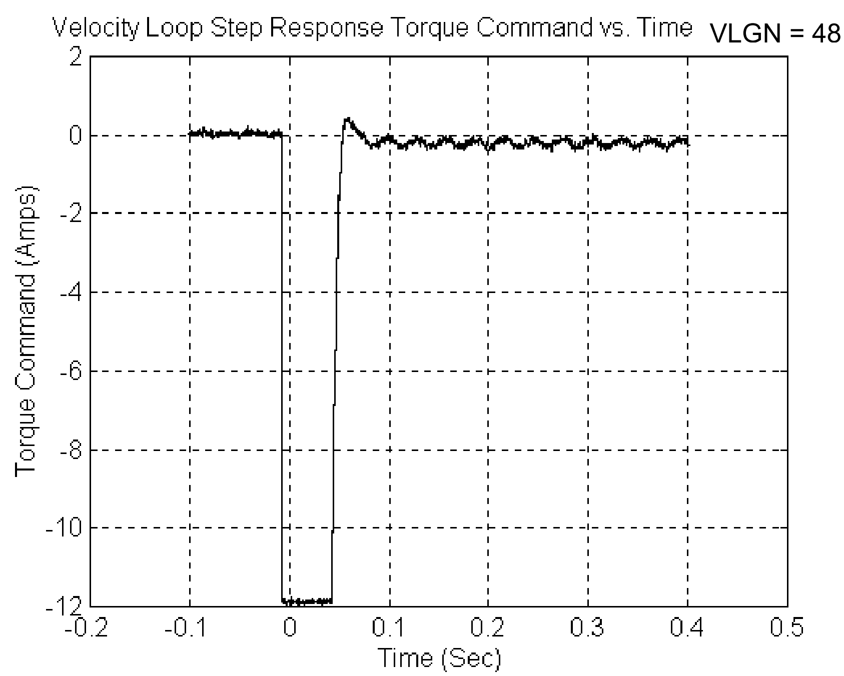


Figure D-7. Velocity Loop Step Response Torque Command vs. Time VLGN = 48

The response shown in Figures D-6 and D-7 is good.

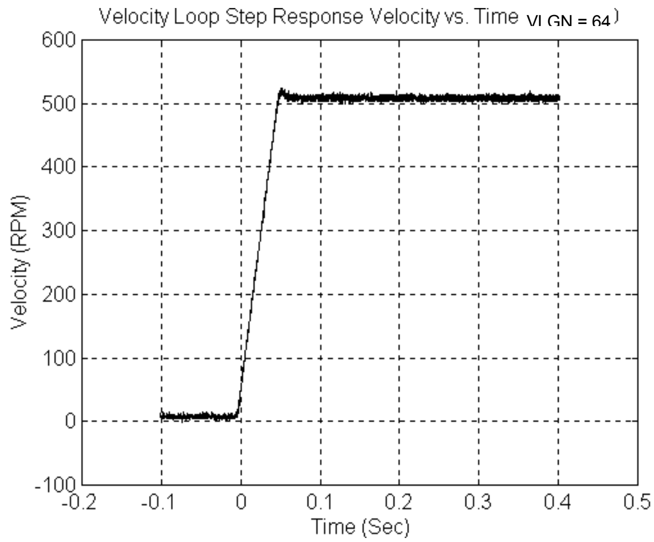


Figure D-8. Velocity Loop Step Response Velocity vs. Time VLGN = 64

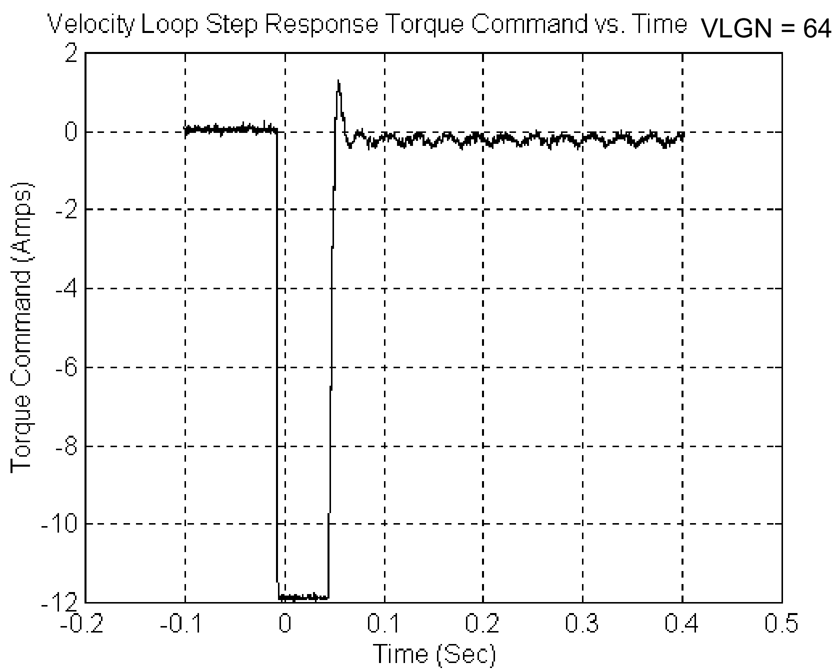


Figure D-9. Velocity Loop Step Response Torque Command vs. Time VLGN = 64

The response shown in Figures D-8 and D-9 is acceptable.

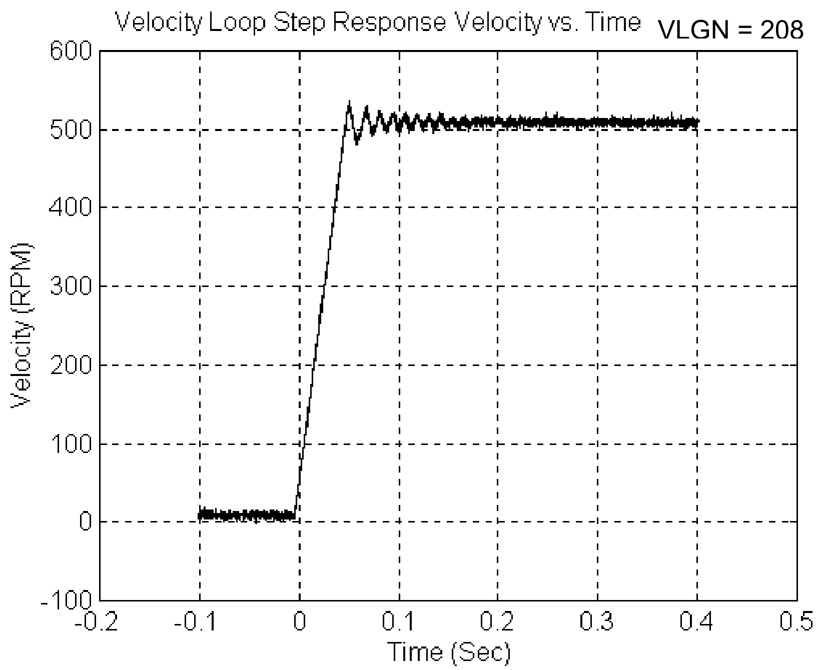


Figure D-10. Velocity Loop Step Response Velocity vs. Time VLGN = 208

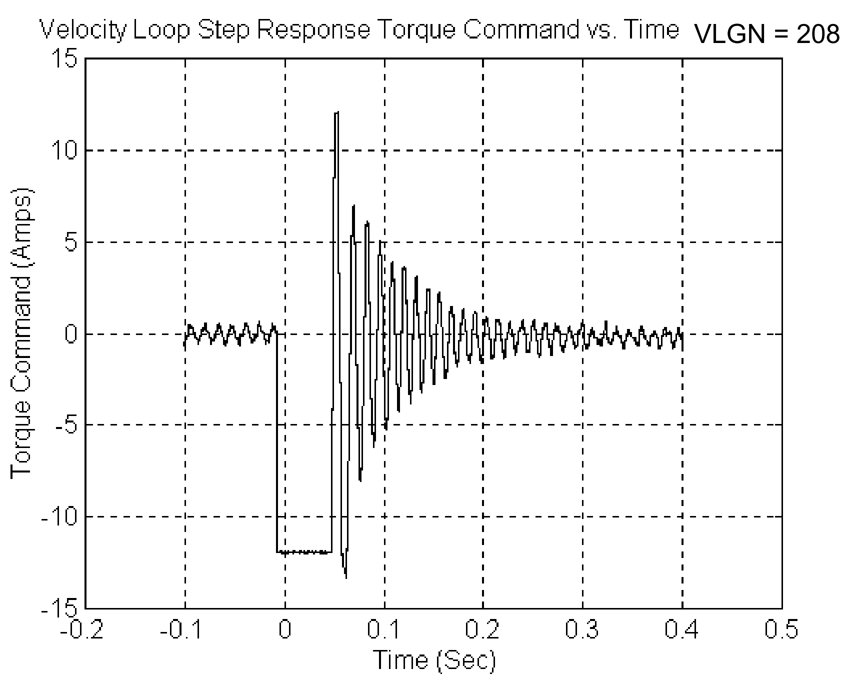


Figure D-11. Velocity Loop Step Response Torque Command vs. Time VLGN = 208

The response shown in Figures D-10 and D-11 is marginally stable and would be unacceptable in many applications. The plots are shown for reference only.

Tuning the Position Loop

The very first step in adjusting the tuning for the position loop is to insure that the velocity loop is stable and has response suitable to the application. Refer to the previous section for methods of setting the velocity loop.

Preliminary Position Loop Settings for Tuning Session.

1. If using Standard Mode control loop settings, set the User Unit and Counts configuration to values appropriate to the mechanical configuration for the axis. See the discussion and examples in the configuration chapter for details.
2. Set the Velocity at 10 Volt value as described in the configuration chapter.
3. Set the Integrator Mode selection to “OFF”.
4. Set the Feed Forward % to zero.
5. Set the Position Error Limit to near maximum value. The maximum is 60,000 (User Units / Counts).

Setting the Position Loop Gain

The position control loop is primarily a “PI” (Proportional, Integral) algorithm with optional Feed Forward. We begin tuning the position loop by setting the proportional gain (Pos Loop TC) to provide a stable response with sufficient gain (bandwidth) to meet the motion profile requirements. By setting the Integrator Mode to “OFF” as requested in the previous section, we create a proportional-only control loop. There are two suggested methods of setting the proportional gain (Pos Loop TC).

Position Loop Proportional Gain Method 1

Calculating the position loop proportional gain assumes that the mechanical design of the machine will have sufficient bandwidth to remain stable and that any resonant frequencies are higher than the bandwidth required by the motion profile.

Terminology

A large mismatch between the load and motor inertia can cause a **RESONANCE** in the system. Resonance is oscillatory behavior caused by mechanical limitations and aggravated by gearing backlash or torsion windup of mechanical members like couplings or shafts. Resonance is eliminated by improving the mechanics, reducing load/motor inertia mismatch or reducing servo gains (reduce performance).

BANDWIDTH is a figure of merit used to compare control system or mechanical performance. As the frequency of command increases, the system response will begin to lag. The bandwidth is defined as the frequency range over which system response (gain) is at least 70% (-3 decibels) of the desired command.

High Bandwidth

- Allows the servo to more accurately reproduce the desired motion
- Allows accurate following of sharp corners in motion paths and high machine cycle rates
- Rejects torque disturbances from mechanics or outside influences improving system accuracy

- Can expose machine resonance, which occur at frequencies near or below the bandwidth

The response of a proportional only system, which is what we set up by setting *Integrator Mode* to "OFF", is an exponential rise. A time constant for an exponential curve represents 68% of the remaining rise. For instance, starting at zero velocity, the response of the position loop to a change in command will require one time constant to reach 68% of the commanded velocity. The second time constant will reduce 68% of the remaining command. Subsequent time constants will reduce 68% of remaining command. For example 100% - 68% (one time constant) = 32%, 32%(68%)=21.8%, 68% (first time constant) + 21.8% (second time constant) = 89.8%. We see that two time constants eliminate 89.8% of the command necessary. Three time constants will account for 96.7% of the rise in command. Four time constants account for 98.9% of the rise. *Typically three time constants are sufficient for most motion applications.*

We can use our knowledge of time constants to predict the required system response. For instance if we know that the fastest acceleration required in our motion profiles must occur within 200 mSec. The 200 mSec response to the change in command will be 98.9% complete in three time constants. Simply dividing the 200 mSec by 3 tells us that a time constant will be about 67 mSec. **The *Pos Loop TC* configuration field represents one time constant in mSec.** In the example above one time constant is 67msec.

Position Loop Proportional Gain Method 2

Similar to the Velocity loop tuning method above. Use an oscilloscope and gradually lower the Pos Loop TC value (increasing gain). Monitor the *Motor Velocity* analog output for performance characteristics are appropriate.

GE Fanuc Start-Up and Tuning Information for Analog Servo Systems

There are two major sections covered;

- Validating Home Switch, Over Travel Inputs, and Motor direction.
- Velocity at 10V, Pos Loop TC, and VEL FF% determination

Analog Mode System Startup Procedures

Startup Procedures

1. Connect the analog motor to the servo amplifier according to the manufacturer's recommendations.
2. Connect the DSM302 **Drive Enable** Relay and **Velocity Command** outputs to the servo amplifier. Connect the position feedback device (Incremental Quadrature Encoder) to the Motion Mate DSM302 encoder inputs.

   Note

   If these connections are incorrect or there is slippage in the coupling to the Feedback Device, an ***Out of Sync*** error condition can occur when motion is commanded.
3. Connect the servo amplifier Ready output (if available) to the DSM302 SRDY input (IN_4). If the servo amplifier does not provide a suitable Ready output, this input to the DSM302 must be connected to 0v. If a **Home** switch is used (24 Vdc), wire it to the correct DSM302 input. The **Home** switch must be wired so that it is ALWAYS ON when the axis is on the negative side of home and ALWAYS OFF when the axis is on the positive side of home.
4. Use the Series 90-30 PLC Configuration Software to set the desired configurable parameters. Store the configuration to the PLC.
5. Turn on the %Q *Enable Drive* bit and place the command code for *Force D/A Output* equal to 0 in the %AQ table. Confirm that the servo amplifier is enabled (the motor should exhibit holding torque). If the motor moves, adjust the amplifier command offset adjustment until the motor stops moving. Note: The %Q *Enable Drive* bit must be maintained ON in order for the *Force D/A Output* command to function.
6. Send the command code for *Force D/A Output* equal to +3200 (+1.0v). Confirm that the motor moves in the desired POSITIVE direction (based on the Motor Dir configuration parameter setting) and the *Actual Velocity* reported in the DSM302 %AI table is POSITIVE. If the motor moves in the wrong direction, consult the servo amplifier manufacturer's instructions for corrective action. The Motor Dir parameter in the Configuration Software can also be used to

swap the positive and negative axis directions. If the motor moves in the POSITIVE direction but the DSM302 reports that *Actual Velocity* is NEGATIVE, then the encoder channel A and channel B inputs must be swapped.

7. Record the actual motor velocity reported by the Motion Mate DSM302 with a 1.0 volt velocity command. Multiply this velocity by 10 and update the *Vel @ 10V* entry in the DSM302 configuration, if necessary. Initially set the *Pos Loop TC* configuration parameter to a high value (typically 100 to 1000 ms).
8. Turn on the %Q *Jog Plus* bit. Confirm that the servo moves in the proper direction and that the *Actual Velocity* reported by the Motion Mate DSM302 in the %AI table matches the configured jog velocity.
9. With the *Drive Enabled* %Q bit ON and no servo motion commanded, adjust the servo drive command offset adjustment for zero *Position Error*. The integrator should be OFF during this process.
10. Check for proper operation of the *Find Home* cycle by momentarily turning on the %Q *Find Home* bit (the *Drive Enabled* %Q bit must also be maintained ON). The axis should move towards the Home Switch at the configured *Find Home Velocity*, then seek the Encoder Marker at the configured *Fnl Home Velocity*. If necessary, adjust the configured velocities and the location of the **Home Switch** for consistent operation. The final **Home Switch** transition MUST occur at least 10 ms before the Encoder Marker Pulse is encountered. The physical location of *Home Position* can then be adjusted by changing the *Home Offset* value in the Configuration Software.
11. Monitor servo performance and use the %Q *Jog Plus* and *Jog Minus* bits to move the analog servo motor in each direction. The *Pos Loop Time Constant* can be temporarily modified by placing the correct command code in the %AQ table. For most systems the *Position Loop Time Constant* can be reduced until some servo instability is noted, then increased to a value approximately 50% higher. Once the correct time constant is determined, the DSM302 configuration should be updated using the Configuration Software. *Velocity Feedforward* can also be set to a non-zero value (typically 90-100 %) for optimum servo response.

Note

For proper servo operation, the Configuration entry for *Vel at 10v* MUST be set to the actual servo velocity (in User Units/sec) caused by a 10 Volt Velocity command to the amplifier.

System Troubleshooting Hints (Analog Mode)

1. The Motion Mate Analog Mode is available in Release 1.40 of the DSM302 or later. The DSM302 requires PLC firmware release 6.50 or greater and Logicmaster 90-30/20/Micro software release 8.02 or greater.
2. The SRDY (servo ready) Input must be applied (connected to 0v) or the Motion Mate DSM302 will not operate.
3. The ENABLE DRIVE %Q control bit must be set continuously to **1** or no motion other than Jog moves will be allowed. If no STOP errors (see Appendix A for error codes) have occurred, the DRIVE ENABLED %I status bit will mirror the state of the ENABLE DRIVE %Q bit. A STOP error will turn off the DRIVE ENABLED output bit even though ENABLE DRIVE input bit is still a **1**. The error condition must be corrected and the CLEAR ERROR %Q control bit turned on for one PLC sweep to re-enable the drive.
4. If the ERROR %I status bit is **1** and the AXIS ENABLED and DRIVE ENABLED %I status bits are **0**, then a STOP error has occurred (Status LED flashing fast). In this state, the DSM302 will not respond to any commands other than the CLEAR ERROR %Q control bit.
5. The CLEAR ERROR %Q control bit uses one-shot action. Each time an error is generated, the bit must be set to **0** then set to **1** for at least one PLC sweep to clear the error.
6. The *CFG OK* LED must be ON or the DSM302 will not respond to PLC commands. If the LED is OFF then a valid DSM302 configuration has not been received from the PLC.

Appendix E Using the APM Motion Programmer Status Screen with the DSM302

The APM Motion Programmer may be used both to program and to monitor the operation of a DSM302 module. While there are NO restrictions regarding programming, there are several issues to be aware of when using the Status screen to monitor the DSM302. The Status screen was designed for use with the Motion Mate APM300 modules IC693APU301 and IC693APU302. The Status screen display will operate with the DSM302 although some screen items have different meanings.

Standard Mode Status Screen

The figure below indicates the Standard mode Status screen items that have a different interpretation for the DSM302.

```
|PROGRM |       |STATUS |       |FILES  |       |SETUP  |FOLDER |UTILTY |PRINT
1       2       3       4       5       6       7       8       9      10

>
+---- Axis 1  2 ---------------- PLC INPUTS ----------- Axis 1 ----- Axis 2 --+
Axis Enabl 1  1 | Ctl01 0 | reserved 0 | Block Number           0            0 |
Pos Valid  0  1 | Ctl02 0 | reserved 0 | Cmd Position          +0           +0 |
Drive Enab 0  0 | Ctl03 0 | reserved 0 | Actual Pos            +0           +0 |
Prog Activ 0  0 | Ctl04 1 | reserved 0 | Posn Error            +0           +0 |
Moving     0  0 | Ctl05 0 | reserved 0 | Strobe Pos            +0           +0 |
In Zone    1  1 | Ctl06 0 | reserved 0 | Cmd Velocity          +0           +0 |
Pos Strobe 0  0 | Ctl07 0 | PLC Ctrl 1 | Actual Veloc          +0           +0 |
Err Limit  0  0 | Ctl08 1 | Error    0 | Status Code 0000   Analog Input +00021|
+---------------+---------+------ PLC OUTPUTS --------------------------------|
Abort Movs 0  0 | Ctl09 0 | Pgm 4    0 |                                      |
Feedhold   0  0 | Ctl10 0 | Pgm 5    0 |         IMMEDIATE COMMANDS           |
Enab Drive 0  0 | Ctl11 0 | Pgm 6    0 |                                      |
Find Home  0  0 | Ctl12 0 | Pgm 7    0 | Axis 1:  LOAD PARAM P1,  +0          |
Jog Plus   0  0 | Pgm 0 0 | Pgm 8    0 |                                      |
Jog Minus  0  0 | Pgm 1 0 | Pgm 9    0 | Axis 2:  SET POSITION    +0          |
Reset Strb 0  0 | Pgm 2 0 | Pgm 10   0 |                                      |
reserved   0  0 | Pgm 3 0 | Clr Err  0 |                                      |
ID: A00001  DISABLED                 MONITOR APM REV : 0.45       LOGIC EQUAL
C:\LM90\DSMTEST.APM                 CTRL: PLC
REPLACE
```

Strobe 1 Position

Pos Strobe 1

Reset Strobe 1 & 2

Non-zero Status Code Displayed in Following Priority Order:
1) Module Status
2) Servo Axis 1 Error
3) Servo Axis 2 Error
4) Aux Axis 3 Error

Aux Axis 3 – Analog Input 1

The following DSM302 items cannot be monitored on the Standard mode Status screen:

%I Data	**%Q Data**	**%AI Data**
CTL13-CTL16	OUT1 control bits	Strobe 2 Position
Position Strobe 2	OUT3 control bits	User Selected Data
Torque Limit		
Servo Ready		
Any Aux Axis 3,4 bits		
New Cfg Received		

Follower Mode Status Screen

The figure below indicates the Follower mode Status screen items that have a different interpretation for the DSM302.

Follower Enabled, Axis 1

Follower Enabled, Axis 2

```
|PROGRM |       |STATUS |       |FILES  |       |SETUP  |FOLDER |UTILTY |PRINT
1       2       3       4       5       6       7       8       9      10

>
+---- Axis 1  2 ---------------- PLC INPUTS ----------- Axis 1 ----- Axis 2 --+
Axis Enabl      | Ctl01    | reserved   | Block Number                         |
Pos Valid       | Ctl02    | Pos Vld3   | Cmd Position                         |
Drive Enab      | Ctl03    | Vel Mis    | Actual Pos                           |
Prog Activ      | Ctl04    | Mis Sign   | Posn Error                           |
Moving          | Ctl05    | reserved   | Enc3 Pos/Vel                         |
In Zone         | Ctl06    | reserved   | Cmd Velocity                         |
Vel Limit       | Ctl07    | PLC Ctrl   | Actual Veloc                         |
Err Limit       | Ctl08    | Error      | Status Code         Analog Input     |
+---------------+----------+------ PLC OUTPUTS --------------------------------|
Abort Movs      | Ctl09    | reserved   |                                      |
reserved        | Ctl10    | reserved   |          IMMEDIATE COMMANDS          |
Enab Drive      | Ctl11    | reserved   |                                      |
Find Home       | Ctl12    | Fnd Hm 3   | Axis 1:                              |
Jog Plus        | resrv    | Int Mstr   |                                      |
Jog Minus       | resrv    | reserved   | Axis 2:                              |
reserved        | resrv    | reserved   |                                      |
En Followr      | resrv    | Clr Err    |                                      |
        NO COMMUNICATIONS           MONITOR APM REV :
C:\LM90\DSMTEST.APM                CTRL: PLC
REPLACE
```

Not supported by DSM302

Execute Program 4-6

Reset Strobe 1

Execute Program 0-3

Sel Int Mstr Axis 1

Sel Int Mstr Axis 2

Non-zero Status Code Displayed in Following Priority Order:

1) Module Status
2) Servo Axis 1 Error
3) Servo Axis 2 Error
4) Aux Axis 3 Error

Axis 3 - Analog Input 1

The following DSM302 items cannot be monitored on the Follower mode Status screen :

%I Data	**%Q Data**	**%AI Data**
CTL13-CTL16	OUT1 Control bits	Strobe 1 Position
Position Strobe 1	OUT3 Control bits	Strobe 2 Position
Position Strobe 2	Reset Strobe 2	User Selected Data
Torque Limit	Execute Program 7-10	
Servo Ready		
Any Aux Axis 3,4 bits		
New Cfg Received		

Appendix F — Updating Firmware in the DSM302

The DSM302 operating firmware is stored in on-board FLASH memory. The firmware update is provided on a floppy disk. The PC Loader utility controls downloading the new firmware from the floppy to the DSM302 FLASH memory. This utility requires an IBM AT/PC compatible computer with at least 640K ram, one floppy drive, MS-DOS® 3.3 (or higher),and one RS-232 serial port. In order to run this utility within an MS-DOS box under Windows® 3.1, Windows® 95 or Windows NT®, the processor should be at least a Pentium™ 133. If not, the computer should be rebooted into MS-DOS mode. This utility functions optimally with a hard drive with at least 1 MB available space.

To Install the New Firmware, Perform the Following Steps:

1. Save or backup any programs or data resident in the module before performing the update function.
2. Place the PLC in STOP/NOIO Mode. (Clear any faults.)
3. Ensure that the module's SNP serial port baud rate is set to 19200 baud.
4. Using a Station Manager to PC cable, IC693CBL316, connect the appropriate serial port of your computer (master) to the DSM302 module to be updated (slave).
5. Insert the supplied firmware upgrade floppy disk into your A: or B: drive.
6. At the C:\> prompt, type A:install (or B:install if your floppy drive is B:). The install program will copy several files to the hard drive then invoke the PC Loader. Install can also be run from the floppy drive directly if there is no hard drive or not enough space on the hard drive. To run from the floppy, type install at the A:\> or B:\> prompt.
7. By default, the PC Loader communicates through the COM1 serial port. A different serial port may be selected by pressing the **F3** key. Press the TAB key to toggle through the selections and press ENTER to accept the displayed choice.
8. From the main menu, press the **F1** key to attach to the DSM302 slave device.
9. Once the slave device is attached, the boot mode menu will appear - press **F1** to enter BOOT MODE and press the **Y** key to confirm the operation. The STAT and CFG LED's on the front of the module should now be flashing in unison.
10. Once in boot mode, press the **F1** key to download the new firmware.

11. Press the **Y** key to confirm the operation. The download time is approximately 4 minutes. If the download fails, refer to *Restarting An Interrupted Firmware Upgrade*.
12. When the download is complete , the PC loader will instruct you to power cycle your module. At this time, power cycle the rack containing the module. If the module is installed in a remote or expansion rack, it is also necessary to power cycle the main rack.
13. Label the unit using the supplied labels.

Restarting an Interrupted Firmware Upgrade

A. Connect all cables as described in step 4 of the procedure above.

B. Power cycle the rack containing the module. If a partial or erroneous download was performed, the module will power up with the STAT and CFG LED's on the module flashing in unison.

C. If you are still running the PC Loader program on your PC, skip to step D; otherwise, follow steps 5 and 6 above.

D. Follow steps 7 and 8 above. Note that you will automatically be placed in BOOT MODE.

E. Follow steps 10 through 13 above.

Appendix G

Answers to Frequently Asked Questions

α and β Series Motors and Amplifiers

What is the meaning of the "IP" ratings associated with the motors?

The degree of motor protection is indicated by a symbol consisting of the two code letters; IP, (International Protection) and two reference numbers indicating the degree of protection. The first number represents the level of protection against physical intrusion. The second number represents the protection against liquid intrusion. This standard does not specify a degree of protection of electrical against mechanical damage, against the risk of explosion, or against conditions such as moisture (i.e. condensation), corrosive vapors, fungus, or vermin.

	First IP Number (Physical Protection)		**Second IP Number (Liquid Protection)**
0	No special protection	**0**	No special protection
1	Solid objects greater than 50 mm	**1**	Dripping water
2	Solid objects greater than 12 mm	**2**	Water falling at an angle to 15 deg.
3	Solid objects greater than 2.5 mm	**3**	Water falling at an angle to 60 deg.
4	Solid objects greater than 1 mm	**4**	Splashed water, any direction
5	Harmful dust deposits, no contact	**5**	Water projected from a nozzle
6	Dust-tight, no contact	**6**	Heavy seas
		7	Effects of immersion
		8	Effects of submersion

EXAMPLE: **IP 55**

An enclosure with this designation is physically protected against harmful dust deposits, with complete protection against contact. The liquid protection is from water projected from a specified type of nozzle.

What is the max cable length and recommended cable length from the controller to the amp and from the amp to the motor?

There are limitations to how far apart the motion control, servo amplifier, and motor can be and still have reliable operation. These are determined by such factors as signal attenuation and resistance losses in the power lines. The DC power to the serial encoder must provide sufficient power level to the encoder.

Motion Controller To Amplifier Separation

Maximum is 50 meters (160 feet)

Amplifier to Motor Separation (Motor Power)

The motor power wiring is normally not a limitation. Wire has to be of sufficient size to not create a great line drop. GE Fanuc cables are sized for 14 meters; if longer cables are required they should be sized for the increased distance. Refer to the servo description manual for details.

Amplifier to Motor Separation (Encoder Feedback)

The normally recommend maximum separation distance is 50 meters (160 feet). Special FANUC approved installations have been used for a maximum amplifier to motor separation of 61 meters (200 feet). For distances beyond the standard 14 meters, a larger gauge cable wire must be provided for the +5 volt DC encoder power to limit the voltage droop to no more than 0.2 volts. The position encoders have a 0.185 amp current requirement. There are techniques that can be used to extend the range using fiber optic cable (requires adapters to convert the signal over and back). The cross sectional size of the serial encoder power wiring can be reduced on long separations by placing a regulated 5 volt DC supply at or near the motor (not recommended for Severe Duty applications).

Can the β Series servomotors be operated on single-phase?

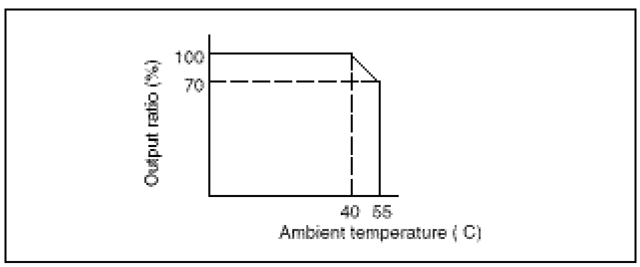
Yes. To use the motors with single–phase power, observe the following:

Power source specification

- Nominal voltage rating: 220 to 240 VAC
- Allowable voltage fluctuation: –15% to +10%
- Frequency: 50/60 Hz
- Allowable frequency fluctuation: ± 2 Hz
- Voltage fluctuation at acceleration/deceleration: 7% or less

Temperature de-rating

Driving the following motors with single-phase power requires temperature de-rating:
ß6/2000 and αC12/2000



When a motor is powered with a single–phase voltage, the lifetime of the related servo amplifier may be decreased due to heat generated if repeated acceleration/deceleration restrictions are not observed. If the ß6/2000 or αC12/2000motor is used for applications that require start/stop cycles of 20 seconds or less, it should be powered with a three–phase voltage.

What are the differences between the α Series and the β Series, Servo Motors and Amplifiers?

Feature	α Series (SVU)	β Series
Torque Range	5.6 – 494 in-lb (0.5 – 40 Nm)	5.6 – 106 in-lb (0.5 – 12 Nm)
Encoder	64K Standard, 1Meg Optional.	32K Standard
Control Power	230 VAC, 1 Phase	24 VDC
Circuit Breaker	Integral	External
Regen Resistor	Integral	External
Encoder Battery	Optional (External)	Optional (External)
Size	Medium	Compact
Motor Poles	8	8
Stator Skewing	Yes	No

What are the benefits of a digital servo system?

- *All Digital System:* All control loops - current, velocity, and position - are closed in the controller. High-speed microprocessors or digital signal processors (DSPs) in the motion controller provide loop update times on the order of 250μs. The servo system can compensate for some machine design limitations yielding faster acceleration/deceleration rates and better response to load changes and wide inertia mismatches.
- *All Digital Servo Command Signals:* A PWM command between the amplifier and motor improves efficiency by varying the on-time of the transistor switches to control motor voltage and current. The GE Fanuc system also provides the benefits of the PWM command from the controller to the drive. The PWM command used in Digital Mode is more immune to noise allowing for increased distances between the controller and the amplifier.
- *High Performance Serial Encoders:* Standard serial encoders built into the motors provide 32K (βSeries) or 64K (α Series) counts per revolution feedback. Serial messages support higher resolutions at high motor velocities than common quadrature encoders and are more immune to noise. An optional battery connection provides absolute position feedback, eliminating the need to re-home the system after a power shutdown.
- *Reduced Tuning and Setup:* There is no need for potentiometer tuning or personality modules; little tuning is required for properly sized drives. All drive parameters are stored in the controller in a standard motor database. Configuration settings are not stored in the drive so it can be replaced with little set-up time. Stored drive and machine parameters are quickly transferred to repeat production machines.

What diagnostic information can be gathered on the β Series Servo amplifier?

The α Series Servo Amplifier can detect many error conditions and provide alarm information. The LED's on the front of the amplifier provide a visual cue to the status of the system by indicating, for example, when the motor and amplifier are ready to function. The ALM LED is turned ON when an alarm condition is detected. When an alarm is detected, power is dropped and the motor is stopped by dynamic braking action. Alarm information is generated in the β Series Servo Amplifier and sent to the attached motion controller. The following alarm conditions are generated in the β Series Servo Amplifier:

- *Overvoltage* - DC link voltage is too high, a larger regeneration resistor needed or AC main voltage too high.
- *Undervoltage* - DC link voltage is too low or loss of AC supply

- *Regen Resistor Overheat* - Average regenerative discharge energy is excessive; reduce load, cycle rate or acceleration speed
- *Over Heat* - Amplifier temperature is too high; reduce load, cycle rate or ambient temperature
- *Fan Failure* - Internal fan failed or jammed
- *Overcurrent* - DC link current is excessive; reduce load or accel/decel rates

Which β motors can you run faster than rated speed and how much faster?

The High Response Vector Function (HRV) allows improved torque at higher speeds in the intermittent operating area. For servos less than 9 Nm continuous, the intermittent torque operating velocity range can be extended approximately 1000 RPM. The continuous operating torque and velocity specification curve remains the same as with the non-HRV motion control. The HRV function is standard on all PowerMotion controllers.

What percentage of the move profile can you spend in the intermittent range of the speed vs. torque curve?

The overload duty characteristic curves are determined based on the motor design temperature restrictions for the heat loss and thermal efficiency of a single motor unit (such as protection based on a thermal protector). The curves are determined by assuming that the motor temperature increases gradually under certain overload conditions and must dissipate waste heat to the ambient temperature.

Therefore, the normal speed torque curves do not apply to the rapid temperature rise occurring, for example, when a servo accelerates and decelerates frequently. However, these duty cycle calculations have limited value in most servo applications. Typically, a RMS torque calculation is required to properly determine if a motor can handle the application requirements. A thermal software function is provided to prevent an abrupt temperature rise in the motor by monitoring for abnormal current in the motor. This function may put a thermal limit to the operation of the motor when it is frequently accelerated and decelerated. Driving units (such as amplifiers) contain their own overheating protection devices. Therefore, note that control may be imposed according to how the equipment is being used. The overload duty curves for the ß Series servos can be found in *the ß Series Servo Motor Descriptions Manual* (GFZ-652332EN).

How to interpret overload duty curves

The servomotor can be driven out of the continuous torque, operating zone, but intermittently. Duty-cycle charts show the Duty (%) and the ”ON” time in which motor can be operated under the given overload conditions. The ON (heat generating) and OFF (heat dissipating) limitations of a motor are calculated using the following procedure.

1. Calculate Torque percent by formula below.

The ON time for a motor can be specified within the torque percent.

$$TMD = \frac{LoadTorque}{ContinuousRatedTorque}$$

2. Calculate “OFF” time by formula.

$$t_F = t_R \times \left(\frac{100}{DutyPercent} - 1 \right)$$

$$t_F = OFF\_Time$$

$$t_R = ON\_Time$$

(Example)

It is desired to operate the ß 2/3000 servomotor under a load of 3 Nm at low speed in a 20 degree C ambient environment for 5 minutes:

Because the continuous rated torque of the ß 2/3000 is 2Nm, we should examine the duty calculations:

Torque percent: TMD = 3/2 = 1.5 (150%)

From the overload duty curve of the ß 2/3000:

Duty percent of the motor when it runs with TMD = 150% for five minutes is:

About 37%

OFF time: tF = 5 * ((100/37) – 1) = 8.5 min

After the motor runs under the above conditions, therefore, it must be kept at a stop for at least 8.5 minutes to dissipate heat build up. Higher ambient temperatures will also change the required rest period.

Does logic have to be put in controller for activation of brake or is the brake controlled automatically from the amplifier?

The α Series and β Series amplifiers do not control the brake function. An external brake relay and appropriately rated 90-volt DC power supply is required. A circuit diagram is provided in the servo manual to illustrate a simple full wave rectifier and transformer to supply the required 90 volts DC.

Can GE Fanuc servo motors be controlled with a controller from another manufacturer?

No, both the α Series and β Series servos are operated via a digital command signal from a GE Fanuc Motion Mate or Power Mate or CNC motion controller.

Do GE Fanuc servos interface with Pulse and Direction command signals from stepper controllers?

No, both the α Series and β Series servos require a PWM command signal from the controller.

What is dynamic braking and how is it different from regenerative braking?

Regenerative Braking

The purpose of a regenerative braking circuit is to dissipate the stored kinetic energy in a rotating, controlled motor when a speed reduction is called for. If regenerative braking were not used, a motor would coast down to a slower speed at a rate based largely upon the momentum of the applied load. This coasting type of slowdown would not be acceptable for applications where a fast, controlled response is desired. Regenerative braking, in contrast, helps reduce motor speed, which gives quick response to a slowdown command from the control. Regenerative braking consists of momentarily treating the motor as a generator and electronically directing its generated current into a power resistor, called a “regeneration” or “discharge” resistor. This removal of

electrical energy from the motor causes a reduction in its speed. GE Fanuc servo amplifiers use regenerative braking in the following speed reduction situations:

- When the control commands a reduction in motor speed.
- When the control commands a motor stop.
- When the control commands a reversal in motor direction.

Dynamic Braking

Dynamic Braking is used in non-controlled stops and involves shorting the motor power lines through low value resistances. This braking is possible due to the fact that in a coasting permanent magnet servomotor, the rotating magnetic fields from its rotor magnets are cutting across the motor's stator (stationary) windings and generating a voltage in them. The dynamic braking circuit conducts the electric current generated in the stator windings into resistors, which convert the current into heat, which dissipates into the air. This electric current creates a torque, which opposes motor rotation causing the motor to rapidly decelerate to a stop. Be aware that this feature cannot be used to positively hold a motor motionless since the braking torque is proportional to motor speed, the braking torque decreases exponentially as the motor decelerates to a stop. If this is required, use the optional holding brake. GE Fanuc servo amplifiers use dynamic braking in the following speed reduction situations:

- When the amplifier power is removed.
- When the amplifier E-stop is asserted.
- When the motion control disables the amplifier.

Can I use the brake in the servomotor to stop my load?

No, the brake is designed to hold the machine when the servo motor control is turned off (a parking brake). It is possible to brake the machine by turning off the brake power at emergency stop such as at the stroke end, but it is not designed to reduce the stop distance. Servo motor brakes have thin friction linings and will be damaged or their service life greatly reduced if used to stop moving loads.

What is β Series Servo with I/O link?

I/O Link is a device network for high-speed serial communications between PLC and CNC/PowerMotion products. The β Series Servo with I/O Link is a slave I/O link device for advanced single axis motion control. Axis control resides in the amplifier. Commands are sent from the host control over the I/O link to the amplifier, which can store up to 32 blocks of motion program. A total of 8 axes can be connected to an I/O link master device. PLC's may support multiple I/O Link master modules.

The β Series Servo with I/O LINK is a dependable and inexpensive option for CNCs requiring additional servos that are unable to support these additional axes. For Series 90-30 applications, it may not be cost effective on low axis count applications, especially with the advent of the less expensive Motion Mate DSM. Below a five axis application the Motion Mate DSM with β Series servos is a more cost effective solution.

When using the β Servo with I/O LINK with a Series 90-30 PLC, the amount of available I/O in the PLC must be considered. 128 discrete inputs (%I) and 128 discrete outputs (%Q) are required for **each** amplifier. The Discrete Command Interface protocol must be used to configure and the load the configuration parameters to the amplifier. The Series 90-30 PLC I/O LINK Master Module can only be configured with %I, %Q (discrete inputs and outputs) or %AI, %AQ (analog inputs and outputs, word format). When sufficient %I and %Q data types are not available on the PLC using %AQ to download some of the bit formatted (discrete) configuration parameters or commands can be difficult. Accessing individual BMI (control and status) bits in a "word" can also be a ladder programmers nightmare. Staying with discrete I/O points may require a more powerful CPU, eliminating any price benefit over the DSM/β servo option.

Coupled with a CNC and the Power Motion Manager software package is where the β Series Servo with I/O LINK is most effective. It may be used as a tool changer or other simple auxiliary axis requiring simple positioning. There already exists parameter settings and information for tool turret configuration.

What is a serial encoder?

A serial encoder transmits positional information from the encoder to the motion control every 250 microseconds. It provides velocity, position and commutation feedback. All digital GE Fanuc motion controllers have a built in serial interface for decoding this positional information. Some benefits of the serial encoder are:

- Higher data rates than quadrature encoder
- Serial messages are not affected by capacitance degradation in the cable which may distort an A/B signals
- Supports high resolution at high speed and is not limited by the upper data rate limitations of quadrature encoders
- Higher resolution than a resolver
- Serial encoders provide a fixed message count per time period
- Battery backed absolute positioning option available
- Increased distance from signal source to the encoder

Motion Mate Controls

What is the difference between coordinated and non-coordinated moves? What is interpolation?

Motion types can be classified into various categories:

- Simple control of position or velocity of a single axis or non-synchronized multiple axes. There may be multiple moves taking place, however there is no point-to-point positional relation between any of the axes. These independent moves constitute the simplest of motion requirements. Any axis that uses acceleration and velocity control is by default a single interpolated axis.
- Motion synchronized to external events, such as registration input. These external events may trigger a change in speed, the start or end of a move, or a position correction.
- Coordinated multi-axis control involves the synchronization of multiple axes. Examples of these types of moves would be interpolation and master/slave. Multi-axes interpolation is the trajectory of motion between one point and another using a group of points existing along a specified path (straight line or arc). In multi-axes interpolated moves all axes will start and stop at the same time and the velocity vector along the path is controlled. Different types of multi-axes interpolation include linear, circular and polar.
- Following motion involves non-program motion of one axis as a linear function of the measured motion, master source. Master sources are typically a quadrature encoder or another axis of motion.
- Complex move profiles such as electronic-camming where motion is performed electronically instead of using mechanical components. Camming is similar to following except the gearing ratio between the master and slave varies (non-linear).

What is the servo update rate of the Motion Mate DSM302 control?

When controlling a GE Fanuc digital AC servo, the DSM302 uses these loop update times:

Motor Current / Torque Loop:	250 microseconds
Motor Velocity Loop:	1 millisecond
Motor Position Loop:	2 milliseconds

How quickly can the DSM302 respond to input events and initiate a move?

The DSM faceplate inputs (motion program CTL bits) are available on the DSM module connectors.

- Strobe data may be captured with a maximum 250 µs position update rate or “jitter” from the serial encoder *plus* 0.5 µs input filter delay.
- Motion program control or branching is a worst case 2 ms position loop delay *plus* the input filter delay (5 ms typical for 24 volt CTL inputs or 0.5 µs input filter delay for 5 volt CTL inputs).
- PLC based functions may receive DSM status (%I and %AI) information (except %AI Actual Velocity which = 128ms) from the DSM asynchronously at 2 ms refresh rate plus 2-4 ms PLC back-plane overhead. The PLC will normally read input data once per sweep. At worst case the PLC may have just missed the DSM update and will need to complete a sweep and begin another to read data from the DSM. The result is that DSM status is available at a 4-6 ms rate or a PLC sweep, whichever is largest.
- PLC commands to the DSM (%Q, %AQ) are normally output to the DSM with one PLC sweep delay. The DSM will process the command in 4-5 ms after receipt.

How many moves can performed simultaneously?

Both the Motion Mate DSM302 and the Motion Mate APM302 allow for two moves to be performed simultaneously. These moves can be independent or synchronized with a block start command.

What processor is in the different Motion Mate controllers?

The Motion Mate APM300 Series uses the NEC V25+ (8.448 MHz) while the Motion Mate DSM302 uses the Intel 386EX (25 MHz) and an Analog Devices DSP.

Can the Motion Mate controls interface to motors from another manufacturer?

The Motion Mate APM300 Series position controller provides a standard +/-10V analog velocity command interface to servos and drives of the customer’s choice, while the Motion Mate DSM302 Releases through 1.30 deliver a single vendor solution to the GE Fanuc digital servos. Release 1.40 of the DSM302 also provides a standard +/-10V analog velocity command interface to servos and drives of the customer’s choice.

Will the MCS300 (Former “Power Mate J”) Series become obsolete?

Yes. The Motion Mate DSM302 supports all of the features presently included in the MCS300 Series at a lower price and power consumption.

When do I need to buy the APM Motion Programmer Software and when can I use program 0.

Program 0 is a short default motion program (20 commands maximum in Standard mode; 9 commands maximum in Follower mode) which is defined in the Logicmaster 90-30 configuration software version 8.02 or higher and downloaded to the DSM302 when the module is initialized by the PLC. Program 0 is created by entering motion commands in an English language text format.

Motion programs 1-10 must be created/edited with version 1.50 or higher of the Motion Mate APM300 Series DOS Motion Programmer software. Program 0 does not support subroutines, the APM Motion Programming Software supports up to 40 subroutines.

What units do we program for velocity, accel, distance, etc.

Both the Motion Mate APM300 and DSM300 Series allow for user scaling of programming units (user units). Both controls operate with:

user units = position units

user units/sec = velocity units

user units/sec/sec = acceleration units

Logicmaster configuration software allows each of the controls to be configured with user define units for position, velocity, and acceleration. The relationship between user units and encoder feedback counts is set by the User Units and Counts configuration values. The range of User Units and Counts is 1-65535. The RATIO of User Units / Counts must be from 8/1 to 1/32. The default scaling is 1 User Unit = 1 Count.

What is the maximum number DSM axes per 90-30 CPU?

1. The *Standard Mode* of the DSM302 requires 40 %AI references per 2 axes.
2. The *Follower Mode* of the DSM302 requires 50 %AI references per 2 axes.
3. 64 %Q and 64 %I references per module.
4. The power supply requirement for DSM302 is 800 ma, 5 Vdc.
5. The DSM302 installs in 1 I/O slot location.
6. The DSM302 requires Logicmaster 8.02 or later for configuration. The DSM will be supported in future releases of CIMPLICITY Control.
7. The DSM302 functions with all Series 90-30 CPU models release 6.01 or later, and may be located in any CPU, local expansion, or remote expansion base plate.
8. Series 90-30 high capacity power supply IC693PWR330 (30 watt) used to calculate rack power capacity to support maximum number of DSM302 modules.
9. The maximum number of DSM302 modules per Series 90-30 system is limited by;
    - Number of available %AI and %QI references supported by CPU model.
    - Maximum number of racks supported by CPU model.
    - **Power Supply capacity per rack. Maximum of 7 modules per rack with PWR330 (30 watt), 3 modules per rack maximum with PWR321 (15 watt).**
    - Maximum number of available module slots per rack.
    - Available PC memory for Logicmaster Configuration Software*.

    The table below represents theoretical maximums and may utilize all of a particular PLC data type. For example the 64 axes (32 modules – 2 axes per module) maximum for the 351- 364 CPUs use all available %I and %Q references in the CPU. The practical number of axes must consider I/O use and sweep impact of the entire system. Logicmaster may not have enough memory (depending on the configuration of the computer it's installed on) to even configure all 32 possible modules for the 351 – 364 CPUs.

Series 90-30 CPU Model	311	313	331	340/ 341	350	351/352/360/ 363/364
Available %AI	64	64	128	1024	2048	See Note 1
Available %QI	512	512	512	512	2048	2048
Total Racks supported	1	1	5	5	8	8
Maximum available I/O slots	5	5/10	49	49	79	79
Maximum Supported Standard Axes	2	2	6	16	Note 2	See Note 2
Maximum Supported Follower Axes	2	2	4	16	Note 2	See Note 2

Note 1: The 351, 352, 360, 363, and 364 CPUs were provided with configurable %AI memory in firmware release 9.0. The range of this configurable memory is 128 to 16,384 words using Logicmaster software. In versions of these CPUs containing earlier firmware, the number of %AI references was fixed at 2048.

Note 2: Theoretical number of 64 axes (32 modules). See discussion in paragraph before this table.

Can I change the end position or velocity of a move already in process?

In the Motion Mate controls the Position Increment immediate command can be used to generate small corrections by a specific number of user units between the axis position and the controller tracking.

The *Feedrate Override* immediate command allows changes to a programmed velocity during program execution from 0% to 120%

It is possible to change a data variable (parameter) used in the motion program if timing considerations are accounted for. The DSM utilizes a look ahead buffer to pre-process motion commands. Only position parameters later in the program sequence than the active command in the buffer may be changed. Since PLC logic is required to change a data value the PLC timing must be accounted for. For more information, see the FAQ entry, “How quickly can the DSM302 respond to input events and initiate a move?”

What dedicated I/O is available for each axis on the DSM302?

The DSM includes positive and negative over travel limits as well as a home input for each axis. There are also two high-speed (250 µs) strobe inputs which capture the current axis position into separate registers. These can be used to support feed-to-registration mark applications for measuring distance between events. The general purpose outputs (CTL9, CTL10, CTL11 and CTL12) are also available for each axis. The new DSM hardware platform has the flexibility to increase the per/axis I/O in future releases.

Are the I/O tables for the DSM302 consistent with the APM300 and MCS300 Series?

. The following table compares the I/O table sizes of the three modules:

Data Type	DSM300	APM301 / MCS301	APM302 / MCS302
%I bits	64	32	32
%Q bits	64	32	32
%AI words	40/50/64	15	28
%AQ words	6/9/12	6	6

The %AI / %AQ lengths can be set during configuration to one of three sizes. The 40/6-default length contains all data for a two-axis STANDARD mode configuration. The next 10 %AI words contain additional information useful for FOLLOWER mode. The last 14 %AI words enable access to additional module analog inputs and outputs. Since the Logicmaster software tends to

concatenate memory space for devices during the PLC system configuration, it is unlikely that most applications will be able to accommodate the larger tables without some changing of I/O references. Because of this, a decision was made to alter the order within each table to provide a logical grouping of I/O on a per/axis basis, which will allow the flexibility necessary for future growth of the DSM product. Ladder programs created for current products that use name references will only require editing of the named reference locations.

Appendix H

DSM302 Revision History

Firmware Release 1.40

Features Introduced in Firmware Release 1.40

DSM Analog Mode

The current DSM module is enhanced in firmware version 1.40 to provide an analog servo interface. The controller provides a ± 10-volt analog velocity command interface signal. The analog interface supports third party motor/amplifier sets that support a ± 10-volt velocity command and a quadrature encoder position feedback interface. The function is currently implemented in the APM product line and allows the DSM to support the same motors and amplifiers as the existing APM controller.

Enhanced Follower Accuracy

The DSM follower feature accuracy is enhanced in DSM firmware version 1.4. In Follower mode, the master axis position is sampled 1.0 mSec to 1.5 mSec prior to using the data within the control. This sampling delay causes the follower slave axis to have a measurable and predictable (based upon motor velocity) following error. To address the delay, a master axis position delay compensation has been added in release 1.4. The delay compensation uses the master axis velocity to correct the follower master position and eliminate the following error due to master axis sampling delay.

Problems Resolved by Firmware Release 1.40

EN3/EN4 LEDs Flash when Performing Slow Jog Function

When performing a slow jog, LEDs EN3 and EN4 incorrectly flashed. The problem was caused by an internal Axis-3/Axis-4 AQ processing error. The result was, that if the DSM configuration used 6 or 9 AQ words, the Force D/A command for Axis-3 and Axis-4 operated incorrectly. This has been corrected in firmware version 1.40, such that the error no longer occurs.

Firmware Release 1.30

Features Introduced in Firmware Release 1.30

Enhanced %AI and %I Processing

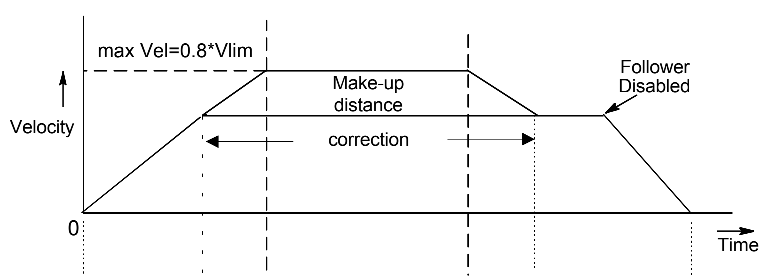
The internal DSM update rate for %I data and %AI data (except for the *Actual Velocity* data) has been increased from once every 10 milliseconds to once every 2 milliseconds. *Actual Velocity* data is updated every 128 milliseconds.

Enhanced Follower Axis Ramp Control

The Follower Axis Ramp Control Function is enhanced in DSM firmware release 1.30. The enhancement improves motion smoothness. Prior to this release, when the follower was active and the master in motion, the acceleration/deceleration control during the make-up correction phase was not controlled by a velocity profile. This could cause unwanted machine shock. The enhanced method uses a trapezoidal velocity profile to address this problem. During the entire make-up correction phase, the acceleration/deceleration is limited to the jog acceleration value. Also during the correction, the velocity is not allowed to exceed the maximum. Appropriate warning/error codes notify the user about abnormal operation (see error table below).

Error Number (Hexadecimal)	Response	Description	Error Type
EA	Status Only	Master velocity greater than 0.8*velocity limit-no distance compensation	Axis
EB	Fast Stop	Error in calculation during ramp-up	Axis
EC	Status Only	Programmed makeup time is not long enough for trapezoidal correction of the makeup distance	Axis
ED	Status Only	Velocity limit violation during ramp	Axis
EE	Status Only	Time limit violation during acceleration sector of the distance correction	Axis

The figure below shows the velocity profile during the follower ramp cycle. Note: The enhanced follower make-up may affect existing applications that use the old follower make-up feature.



Problems Resolved by Firmware Release 1.30

AQ Command 30h Causes Module to "Crash"

Issuing an AQ command of 30h (Set Internal Master Velocity in Follower), to a value larger that 32,767,999 generated a fatal error (NMI generated/watchdog timeout). This was fixed in firmware version 1.30. Error checking was added to generate a warning when values outside the valid range (-1,000,000 … +1,000,000 counts/sec) are entered. If values outside the valid range are entered, the command is ignored and error code01E9 (for Axis 1) or 02E9 (for Axis 2) is reported.

IPI for Release 1.20 Contained Inaccurate Encoder Position Resolution Data

The description for the Enhanced Position Loop Resolution function description was not clear. The description was improved in the IPI for release 1.30.

Turn off "CONFIG LED" when Flashing Error Code

A fatal error did not turn off the "CONFIG LED" when a fatal error code occurs. Firmware version 1.30 was updated to turn off the "CONFIG LED" when a fatal error code is issued.

Jog Vel-User Units-Counts Configuration Value Causes Module to "Crash"

Configuring the module for User Units to Counts ratio of greater than 1:3, and Jog Vel = 8,388,607 caused a fatal error (NMI generated/watchdog timeout). Firmware version 1.30 was enhanced to internally limit the Jog Velocity to 1,000,000 (count/sec). If the module is configured to a number greater than 1,000,000 (counts/sec), the module uses the maximum jog velocity of 1,000,000 (count/sec).

Follower Deceleration Ramp Reentry after Drive off/on

In follower mode, if follower is disabled and then the drive is disabled, motion will stop. However, the module continues to calculate the deceleration ramp while the follower is disabled. If the deceleration ramp has not reached zero prior to the drive being re-enabled, the module will issue a velocity command corresponding to the current deceleration ramp value and complete the deceleration ramp. This was corrected in firmware version 1.30, such that the deceleration ramp is not re-entered.

Firmware Release 1.20

Features Introduced in Release 1.20

Expanded Follower A/B Ratio

The A/B slave/master follower ratio has been expanded from the original range of 32:1 to 1:32 to a range that supports 32:1 to 1:10,000. Existing AQ command 2Dh can be used to specify an expanded range at runtime. Specifying the expanded range (ratio greater than 1:32) at configuration time requires release 9.0 or higher of the MS-DOS Programming Software, or Windows-based Programming Software release 2.11 or higher.

Enhanced Position Loop Resolution

Enhanced position loop resolution, at the expense of maximum supported motor velocity, was added to the product in firmware release 1.20. Prior to this release, a non-configurable position loop resolution of 8192 counts per encoder revolution was provided. The table below describes the various selections now supported, along with the maximum supported motor velocity for each setting. Note that the configuration data is specified by entering a value of “1” (to select parameter 1) in the “Tuning Par1” or “Tuning Par2” field of the Axis-1 screen for axis-1 or Axis-2 screen for axis-2. The appropriate resolution setting value (0..3) is then entered in the corresponding “Tuning Dat1” or “Tuning Dat2” field in the Axis-1 screen for axis-1 or Axis-2 screen for axis-2.

Enhanced Position Loop Resolution Selections Supported

Encoder Resolution (in Counts per Revolution)	Maximum Motor Velocity (Revolutions per Minute)	Configuration Data	
		Parameter #	Value
8192 cts/rev	4400 rpm[1,2]	1	0
16384 cts/rev	3662 rpm[2]	1	1
32768 cts/rev	1831 rpm	1	2
65536 cts/rev	915 rpm	1	3

[1] Default Setting.
[2] Some motors are restricted to a lower maximum rpm rating.

Problems Resolved by Firmware Release 1.20

Input IN4_C Does Not Function As Described

The on/off state of the IN4_C input (see GFK-1464, *DSM302 for IC693 PLCs User's Manual*, chapter 5, "DSM302 to PLC Interface," for details) was inverted from what was documented. This was corrected in firmware version 1.20, such that the on/off state of the input matches the documentation.

Firmware Reports D6 Error Sporadically during Normal Operation

Previous firmware contains an error that causes a D6 error to be reported incorrectly during normal operation. The nature of the firmware error also causes errors D3-D9 to be reported incorrectly. These firmware errors were fixed in version 1.20.

Documentation Issues in *DSM302 for IC693 PLCs User's Manual*, Resolved by Revision A Release

The following table outlines the previously identified documentation issues that have been resolved by the revision A release of GFK-1464*, DSM302 for IC693 PLCs User's Manual:*

Issues Resolved by Revision A of GFK-1464

Documentation Issue	Location	Description/Resolution
PCR Connectors Mislabeled in Pin Configuration Diagrams	Chapter 2	Two different connector pin configurations for the emergency stop JX5 connector on the β Series servo amplifier were shown with the labels (HIROSE and HONDA) incorrectly reversed. The labels were switched.
Grounding Bars and Clamps Need to Be Documented	Chapter 3	The "I/O Cable Grounding" section did not specify the part numbers for the grounding bars and clamps needed for proper installation. These have been added to the manual.
Quadrature Specs. Undocumented	Chapter 3	The technical specifications for quadrature devices used as a follower master axis were not documented. These have been added to the manual.
Incorrect Part Numbers Listed for Terminal Block Connection Cables	Chapter 3	The incorrect part numbers were given for the Terminal Board Connection cables. The incorrect part numbers listed were IC800CBL324 and IC800CBL325. These part numbers should have been IC693CBL324 and IC693CBL325, respectively, and have been corrected.
Input IN4_C Wiring Not Described	Chapter 3	The appropriate wiring for input IN4_C was not described. The description has been added.
Final Home Velocity Valid Command Range Incorrectly Specified	Chapter 4	The "Configuring the DSM302" section incorrectly specified the valid range as 1..8,388,607. The range is actually 1..65535 and has been corrected in the manual.
Select Return Data Command Incorrectly Described	Chapter 5	Section incorrectly stated that information is returned in the *Commanded Position* %AI word for each axis. Data is actually returned via the *User Selected Data* %AI word for each axis. Text has been corrected.
Wrong Graphic Used for Velocity Profile	Chapter 7, Figure 7-6	Figure showed an expected velocity profile for a program example describing "hanging" the DSM302 when distance runs out. The incorrect graphic that was included has been replaced with the correct one.
Error Code 35 Not Documented	Appendix B	Axis status error code 35 was not documented. This error is now correctly described as a "find home while follower enabled" error.
Wire Size, Screw Torque and MOVs for Terminal Blocks Not Specified	Appendix E	The wire size, screw torque and MOV descriptions for terminal block assemblies IC693ACC335 and IC693ACC336 were not specified. These descriptions have been added.

Firmware Release 1.10

Features Introduced in Firmware Release 1.10

HV Motors

Support for the following three HV motors was added in firmware release 1.10:

Motor Model	Motor Type Code
α12HV/3000	3
α22HV/3000	4
α30HV/3000	5

Set Aux Axis 3 Position Command Enhancement

The *Set Aux Axis 3 Position* command was enhanced to execute regardless of the axis' current velocity. In firmware version 1.00, the command would be ignored and a 0X52 axis status error reported if the axis' velocity exceeded 128 counts per second.

Problems Resolved by Firmware Release 1.10

Encoder 3 Home Position

During a *Find Home* cycle on the Aux 3 axis, the Encoder 3 home position was not registered correctly in firmware version 1.00 when the encoder marker pulse was sensed. This was corrected in firmware version 1.10. Note that this problem only pertained to Follower mode operation.

Module Sometimes Halted Responding to COMMREQ Commands

When the DSM module was receiving a high rate of COMMREQ commands from the PLC CPU and was simultaneously executing a series of short move commands, the module would sometimes halt execution and flash a 6-count/ 7-count error code on its STAT LED (see Appendix B of GFK-1464, *DSM302 for IC693 PLCs User's Manual,* for details). This timing problem was corrected in firmware version 1.10.

Index

%

5

A

R

S